Introduction

to the Apple® -Digital

Network Environment

Apple Computer, Inc. and Digital Equipment Corporation PROVIDE THIS PUBLICATION "AS-IS" AND make no warranty or representation, either express or implied, with respect to this publication, its quality, performance, merchantability or fitness for a particular purpose. You are assuming the entire risk as to its quality and performance.

APPLE COMPUTER, INC. AND DIGITAL EQUIPMENT CORPORATION MAY MAKE MUTUALLY AGREED TO CHANGES TO THIS PUBLICATION AT ANY TIME AND WITHOUT NOTICE.

Mention of products herein is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple Computer, Inc. and Digital Equipment Corporation assume no responsibility with regard to the use of the specifications, protocols, or other information herein.

References in this publication to Apple Computer, Inc. products, programs, or services or Digital Equipment Corporation products, programs, or services do not imply the availability of such products, programs, or services.

The content of this publication has been jointly developed by Apple Computer, Inc. and Digital Equipment Corporation.

Printed in the United States of America.

Contents

1 The Apple-Digital Agreement

In January 1988, Apple Computer, Inc. and Digital Equipment Corporation signed a Development Efforts Agreement (the Apple-Digital agreement). This R&D effort provides a framework for integrating Apple® and Digital computing environments. Surpassing mere Macintosh®-to-VAX™ connectivity, the joint development program seeks to create a common information environment for a new class of distributed computing applications.

The goals of the agreement are to develop

- an architectural foundation for connectivity and interoperability between environments
- open interfaces and tools that enable developers to build distributed applications
- a core set of user applications and network services for sharing resources and accessing information

Apple and Digital worked in close collaboration toward these goals by developing the key architectures and enabling technologies, and by continuing to work with innovative commercial developers. The agreement yields DEC LanWORKS™ Software for Macintosh® Computers, which provides a tightly integrated Apple-Digital network environment with associated support and services.

The challenge

Corporations and institutions face the need for people to work together, making the best of their unique talents and skills. Because tasks are shared, people depend on the ability to share information easily and to work together effectively with workers down the hall or around the world. To meet this need, information systems (I/S) have evolved that meet the needs of the enterprise, and individuals have embraced personal computing for personal empowerment. Unfortunately, these computing environments evolved independently. In addition, I/S has implemented a broad range of multivendor tools to better manage this information environment —databases, local area and enterprise networks, and specialized applications to name a few. The challenge for I/S organizations has been to reconcile the personal computing environment with large-system computing. The goal of I/S has been to provide a framework for integrating the best of both of these environments, bringing enterprise-wide integration in a manner that is easy and productive for users.

The solution

Digital’s Network Application Support (NAS) is a comprehensive set of software (applications programming interfaces, tool kits, and products) that enables applications integration across a distributed multivendor computing environment. NAS provides application access, communication services, and information/resource sharing to Macintosh computers and AppleTalk® networks. NAS also supports other environments, which include ULTRIX™ and VMS™ workstations, VT™ terminals, X terminals, and IBM PCs and com-patibles. Digital implements NAS services on a network foundation of DECnet™/OSI and TCP/IP with gateway access to SNA.

Apple’s design center is the user. As such, Apple has developed an integration strategy focused on extending a user’s reach into the larger computing system environments while retaining the intuitive, consistent Macintosh user interface. Apple is committed to bringing integration tools into four strategic network environments: Digital, IBM SNA, TCP/IP, and OSI multivendor systems.

Therefore, the Apple-Digital development agreement is a complementary component of Apple’s multivendor AppleTalk integration strategy and Digital’s Network Application Support.

The joint development effort seeks to go beyond the mere creation of “point-products.” A careful long-term perspective is taken to ensure that:

- The results of the development effort meld two network architectures, develop enabling technologies, developer tools, and a wide range of functionality for Macintosh-to-VAX integration. The companies have been involved in a complex task with many components developed in a manner that is consistent with each company’s long-range product strategies and in a way that protects each company’s customer investments for the future.
- The joint development agreement covers three major areas of development: network connectivity and interoperability, distributed applications, and business communications services. Apple and Digital will provide the standard connectivity platform with a standardized framework and open interfaces so that developers get a long-term, stable network development platform. This environment also increases the compatibility of third-party products.
- The two companies work to enhance the support and service offerings required for multivendor networks.

Figure 1-1 shows the primary means by which Apple and Digital connect the Macintosh and VAX™ network environments. This approach offers the advantages of computers and networks from both Apple and Digital, yet it preserves the investment that Apple and Digital customers have in their current systems.

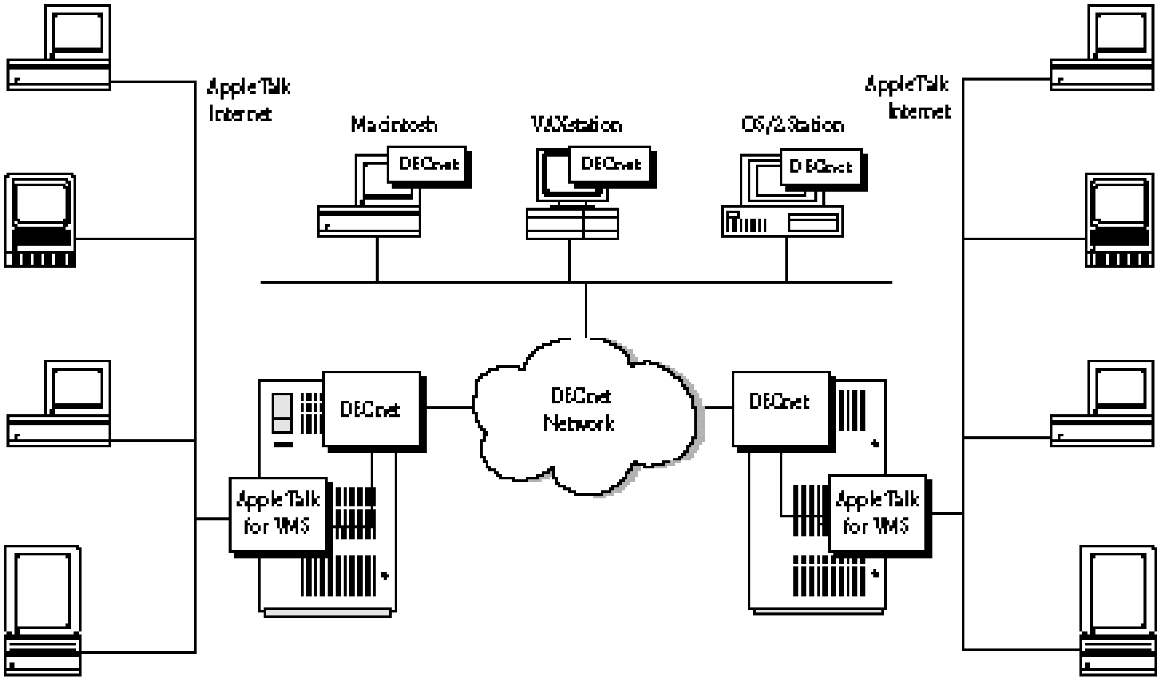


Figure 1-1 Apple-Digital network connectivity

The benefits

One of the primary goals of the joint development agreement is to improve and standardize the technology infrastructure for developers of networking services and user applications. This standardized infrastructure will provide reliable connectivity, set the stage for a new generation of networked applications, and provide for access to VMS applications by means of a consistent Macintosh user interface.

A standardized technology infrastructure and base set of network services (applications)

- protect investment in the multivendor environment
- assure major vendor support
- provide a stable development environment that frees management information systems (MIS) developers to concentrate on applications for the company’s competitive advantage

With Macintosh and AppleTalk integration, VAX system users get

- an expanded choice of desktop devices to access VMS applications and DECnet services
- the ability to work with Macintosh-produced documents and files
- the ability to share AppleTalk resources (such as LaserWriter® printers) with other Macintosh and MS-DOS users

With VAX and DECnet integration, Macintosh users get

- seamless access to a powerful server platform and extensive computing power
- the ability to exploit network investments by reaching farther by way of the DECnet/OSI wide area network (WAN)
- access to corporate data management services and security
- the ability to share information and resources with MS-DOS, OS/2, VMS, and ULTRIX users on the same network

The joint development agreement delivers complementary network computing in a manageable structure for empowerment of the user. The Macintosh provides individuals and workgroups an uncomplicated, graphic view of the world of information. It lets them work intuitively, see more aspects of their businesses by representing information in multimedia formats, and transparently reach beyond the desktop to make use of resources on the network. The VAX provides organizations a cohesive structure for managing personal-computer information and resources. Because this environment is based on solid architectures and standards, and supported by Digital and Apple, corporate investments are protected as well.

Multivendor computing requires connectivity between different systems. The environment for connectivity between the AppleTalk and DECnet/OSI network systems is based on

- the Macintosh computer’s communications interface (known as the Macintosh Communications Toolbox)
- AppleTalk for VMS (the protocol stack and router)
- the AppleTalk-to-DECnet transport gateway
- DECnet for the Macintosh

These components, along with Apple’s AppleTalk software, which is standard on every Macintosh computer, and Digital’s DECnet/OSI network products, are the main components of Apple-Digital connectivity. In addition, both companies support TCP/IP-based communications. Figure 2-1 illustrates Apple-Digital connectivity.

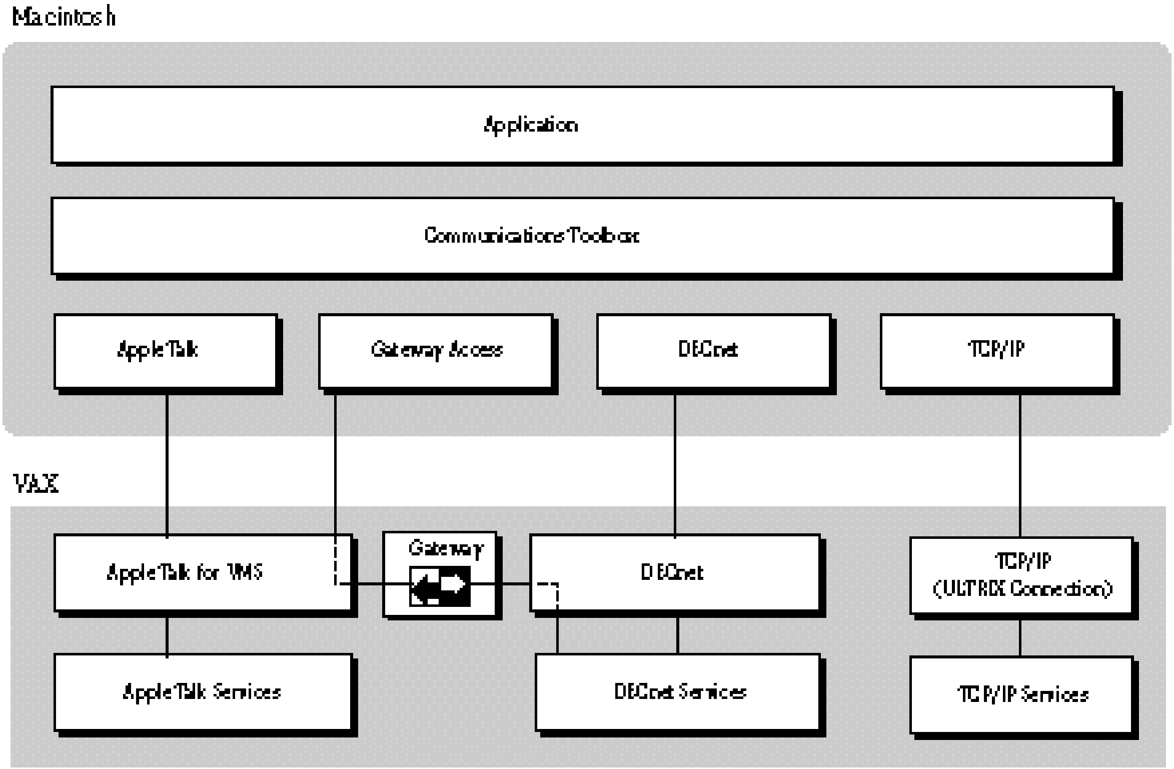


Figure 2-1 Overview of Apple-Digital connectivity

The sections that follow introduce the components that interconnect Apple and Digital networks: the Macintosh Communications Toolbox, AppleTalk, AppleTalk for VMS, DECnet, the AppleTalk-to-DECnet transport gateway, and network management capabilities.

Macintosh Communications Toolbox

The Macintosh Communications Toolbox is Apple’s strategic communications development platform designed to support multivendor connectivity for Macintosh computers in DECnet/OSI, AppleTalk, and other environments. Figure 2-2 shows the basic architecture of the Communications Toolbox.

The Communications Toolbox is system software that provides Macintosh applications with standard access to communication services including data connections, terminal emulation, and file transfer protocols. An extension of the Macintosh Toolbox, it consists of *managers* and associated *tools* that work in concert to provide applications with standard communications functions. The four primary managers are the Connection Manager, the Terminal Manager, the File Transfer Manager, and the Communications Resource Manager.

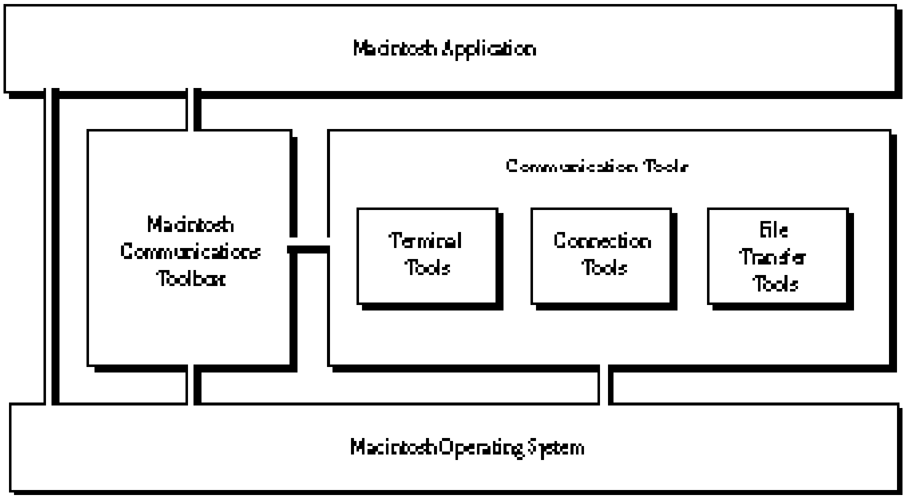


Figure 2-2 High-level architectural diagram

How the Macintosh Communications Toolbox works

Apple, Digital, and third-party developers provide tools that implement particular communications functions—for example, a modem connection tool or, for high-performance Ethernet connectivity, Digital’s LAT™ (local area transport) technology.

All communications applications written according to the Macintosh Communications Toolbox standard provide users with a consistent Macintosh user interface, thereby making communications capabilities much easier to use. Developers benefit from the standard applications programming interfaces (API) of the Communications Toolbox in the following ways:

- Application developers have ready-to-use tools for implementing communications services in any program.

- Hardware developers can register their NuBus™ communications cards (for example, serial or modem cards) with the Communications Toolbox, making them accessible from both applications and tools in a standard fashion.
- Specialized communications software developers are able to focus on their expertise by creating communications tools that provide a particular communication service to any Communications Toolbox-aware application.
- The APIs offer transport independence.

Benefits of the Macintosh Communications Toolbox

The Macintosh Communications Toolbox has an open, modular, extensible architecture that

- enables developers to create applications more easily since they can be developed independent of particular protocols, by
  - providing applications with protocol-independent access to a byte-stream data connection
  - allowing execution-time binding of tools
  - supporting any registered NuBus communications card
- provides consistent network/service access and configuration interfaces to users, by
  - allowing users to easily configure and select communication services
  - allowing users to add tools to and remove them from the System Folder and make them available to any Communications Toolbox application instantly, without restarting the system or application

The Communications Toolbox represents a new level of system software support for the Macintosh networking and communications environment.

AppleTalk

AppleTalk refers to Apple’s network protocols and those Apple and Apple developer products that comprise the AppleTalk network system—a full seven-layer media-independent network system. AppleTalk offers “plug and play” connectivity, thereby extending the power and ease of use of Apple computers to a distributed environment. Introduced in 1985, AppleTalk has been a standard part of Macintosh system software, is now widely accepted in the industry, and has an installed base of over two million computers.

AppleTalk provides and supports many network services, such as

- AppleShare® file service and print spooling for Macintosh and Apple II computers, and for computers running the MS-DOS operating system
- PostScript-based printing on Apple’s LaserWriter printers
- electronic mail services
- internetwork routing
- dial-up network access
- network management
- interapplication communications

The AppleTalk architecture, an open protocol that is well defined and fully documented by Apple, is compliant with the International Standards Organization network model. AppleTalk provides networking services that let applications and system software of different AppleTalk network nodes interact, both with each other and with a variety of servers. Figure 2-3 shows the AppleTalk communications architecture (items shaded in gray are not currently implemented in AppleTalk for VMS).

The AppleTalk network system offers a wide choice in cabling and network data-links including Apple’s economical LocalTalk®, which is built into every Macintosh, and industry-standard Ethernet (802.3) and token ring (802.5) networks. AppleTalk is supported on a variety of computer platforms including VMS, UNIX, MS-DOS, OS/2, NetWare, and other environments.

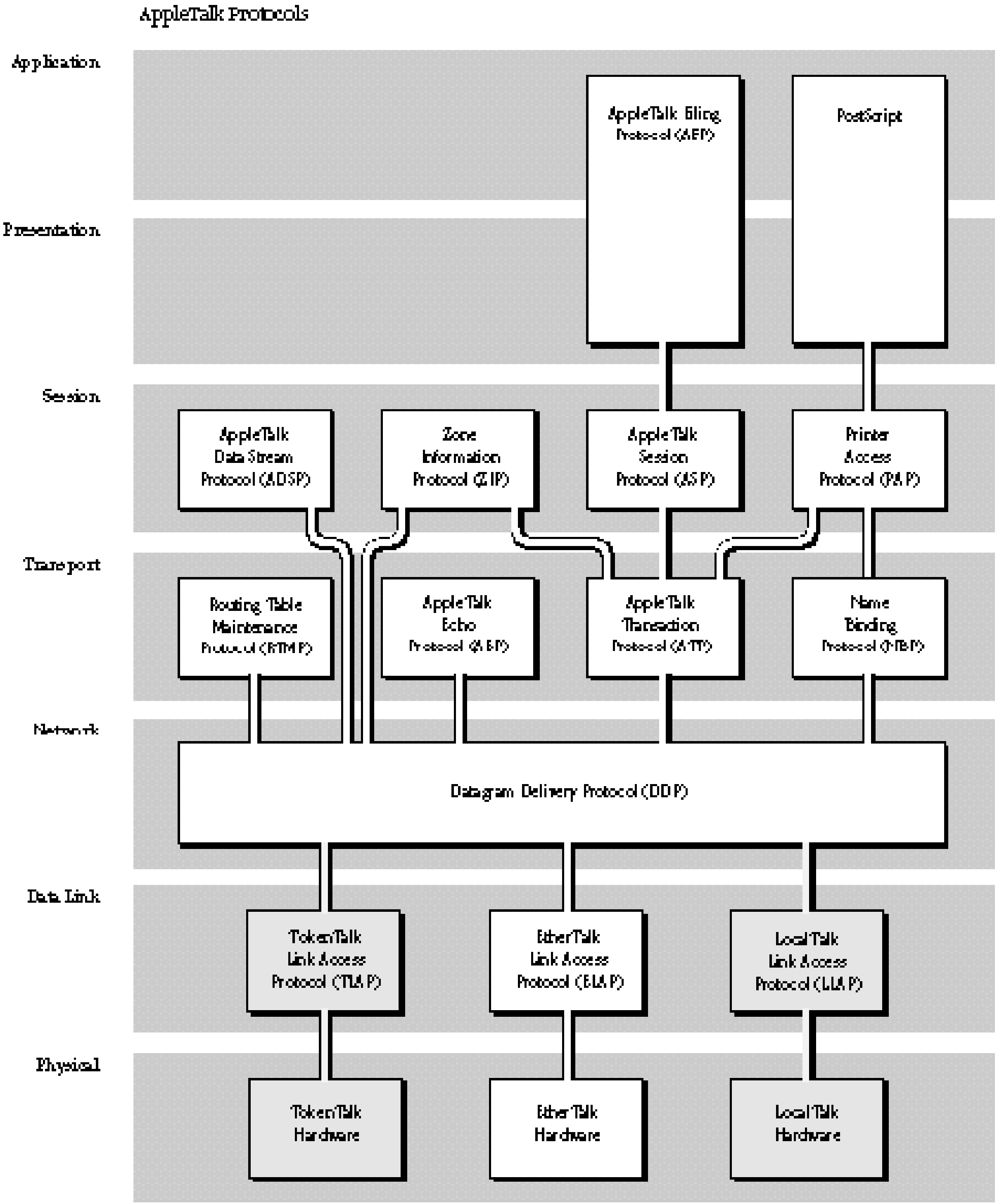

AppleTalk Protocols
Application
Presentation
Session
Transport
Network
Data Link
Physical
AppleTalk Filing Protocol (AFP)
PostScript
AppleTalk Data Stream Protocol (ADSP)
Zone Information Protocol (ZIP)
AppleTalk Session Protocol (ASP)
Printer Access Protocol (PAP)
Routing Table Maintenance Protocol (RTMP)
AppleTalk Echo Protocol (AEP)
AppleTalk Transaction Protocol (ATP)
Name Binding Protocol (NBP)
Datagram Delivery Protocol (DDP)
TokenTalk Link Access Protocol (TLAP)
EtherTalk Link Access Protocol (ELAP)
LocalTalk Link Access Protocol (LLAP)
TokenTalk Hardware
EtherTalk Hardware
LocalTalk Hardware

Figure 2-3 AppleTalk communications architecture

AppleTalk Phase 2

In June 1989, Apple introduced an extension to the AppleTalk protocols called AppleTalk Phase 2. This version of AppleTalk was enhanced to make large-scale local area internetworking more efficient and easier to manage. At the same time, AppleTalk Phase 2 is true to the original AppleTalk design center and remains the easiest network system to use.

Benefits of AppleTalk Phase 2

AppleTalk Phase 2 provides the following benefits:

Extended addressing AppleTalk addressing comprises an 8-bit node ID and a 16-bit network ID. AppleTalk Phase 2 provides for assignment of a range of network numbers to a single network segment, which allows more than 16 million AppleTalk nodes to be addressed on a single network. This addressing feature was added to accommodate extended networks that use bridges and routers to interconnect network segments, and brings to AppleTalk the flexibility to support large mixed internetworks.

Efficient internetwork routing AppleTalk Phase 2 provides more efficient routing by giving AppleTalk nodes the ability to determine the best route to take to any particular destination on the internetwork. This reduces retransmissions of internetwork-bound traffic and generally improves the performance of internetwork communications.

Reduced network traffic One of the main benefits of AppleTalk Phase 2 is the reduction of network traffic. This reduction is accomplished in three areas:

- *Zone multicast.* This feature takes advantage of the multicast ability of Ethernet and token ring networks to isolate broadcast traffic associated with name lookups to any preselected group of AppleTalk nodes. Such a group is called a zone. Because the multicast address is a data-link attribute, broadcasts are effectively isolated to the zone and do not interrupt other AppleTalk nodes that do not belong to that zone or non-AppleTalk nodes on the network.

- *Split-horizon RTMP broadcast.* Each AppleTalk router maintains a table of all networks on the internet, which it uses to deduce the routes by which to send a packet to its destination network. AppleTalk routers broadcast these tables to other routers on their network using a protocol called the Routing Table Maintenance Protocol (RTMP) to keep the tables of all AppleTalk routers reconciled. The split-horizon feature in AppleTalk Phase 2 routers greatly reduces RTMP traffic on backbone networks by forwarding RTMP information only to routers on networks that have not received that information.
- *AppleTalk multicast.* This feature takes advantage of the multicast ability of Ethernet and token ring networks to isolate all AppleTalk traffic from non-AppleTalk nodes. It isolates AppleTalk traffic not targeted to a specific zone, such as RTMP broadcasts, so that they do not interrupt non-AppleTalk nodes on the network.

Support for network standards AppleTalk Phase 2 brings support for networking standards such as 802.3 Ethernet and 802.5 token ring. Using the AppleTalk routers, these network standards can be interconnected with Apple's LocalTalk to provide transparent access to services anywhere on the internetwork.

Upgrading to AppleTalk Phase 2

All AppleTalk users should upgrade to AppleTalk Phase 2. AppleTalk Phase 2 is the future for networking Apple computers and will replace the original AppleTalk. For example, AppleTalk for VMS version 3.0 (described in the next section) supports only AppleTalk Phase 2. Existing AppleTalk Phase 1 devices can communicate with this version of AppleTalk for VMS through the AppleTalk Internet Router but should be upgraded to avoid this extra step.

Upgrading to AppleTalk Phase 2 is a matter of installing Phase 2 software for Apple products. Apple developers offer AppleTalk Phase 2 upgrades for their products as well. A migration utility is available with the AppleTalk Internet Router for large AppleTalk internetworks that plan to upgrade in steps as opposed to all at once.

AppleTalk Phase 2 gives Apple users an intuitive view of network services and ensures that these services work in a consistent manner, though they may reside on different computing platforms.

AppleTalk for VMS

AppleTalk for VMS is Apple's implementation of the AppleTalk network protocols on Digital's VMS operating system. With AppleTalk for VMS, a VAX computer system can participate in an AppleTalk internet. Likewise, any computer on the AppleTalk internet can access the VMS environment of a VAX computer. Apple and Digital have chosen AppleTalk for VMS and its integration with DECnet as the primary way to interconnect the products they develop under the Apple-Digital agreement. Thus, using AppleTalk for VMS as a base, a developer can build distributed applications across Macintosh, Apple II, MS-DOS, and VMS systems. Figure 2-4 illustrates connectivity by means of AppleTalk for VMS.

With AppleTalk for VMS, VMS system-based applications and services can appear just like other AppleTalk services. These VMS system-based services register their names on the AppleTalk internet in the same way that AppleShare servers and LaserWriter printers register their names. Macintosh users can find VMS system-based services through the Chooser or through application-specific interfaces.

AppleTalk for VMS has the following features:

- AppleTalk Phase 2 routing
- enhanced performance by running in the VMS kernel
- enhanced tunneling capabilities
- support for the AppleTalk ADSP to DECnet NSP transport gateway
- a more complete and easier configuration and management utility
- a simple, well documented API

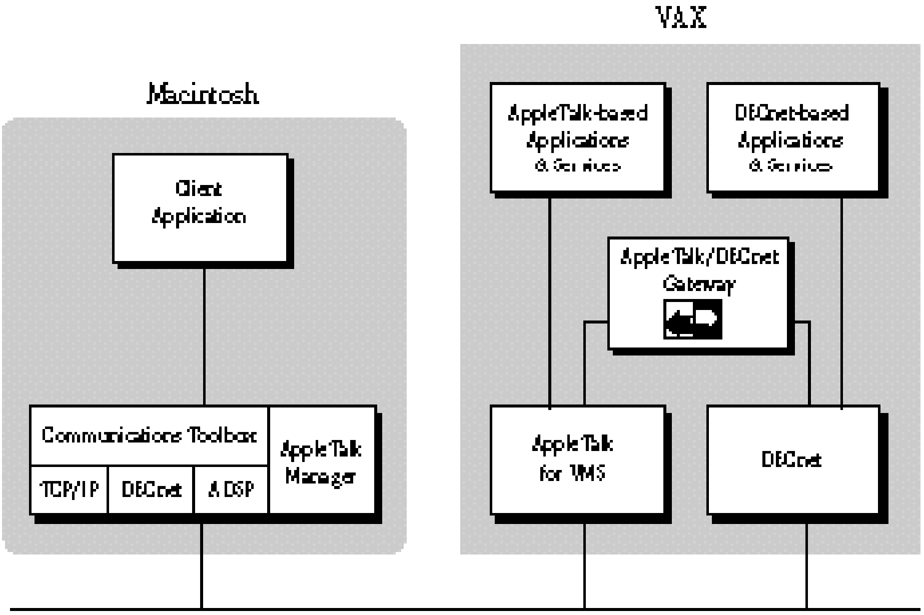


Figure 2-4 Overview of Apple-Digital connectivity with AppleTalk for VMS

AppleTalk for VMS router

In the AppleTalk network architecture, routers forward datagrams between separate AppleTalk networks. With AppleTalk for VMS, a VAX computer can become a full-function AppleTalk router, providing internet routing, zone information management, routing table maintenance, and management of name-binding requests.

An AppleTalk for VMS router has multiple ports, so that it can simultaneously route datagrams over different physical communication channels. For example, a router can route datagrams over multiple Ethernet local area networks or over DECnet/OSI wide area networks. By encapsulating AppleTalk datagrams into a DECnet packet and routing this packet through DECnet/OSI networks—a technique called *tunneling*—the AppleTalk for VMS router can interconnect AppleTalk internets separated by great distances. Figure 2-5 illustrates AppleTalk for VMS routing.

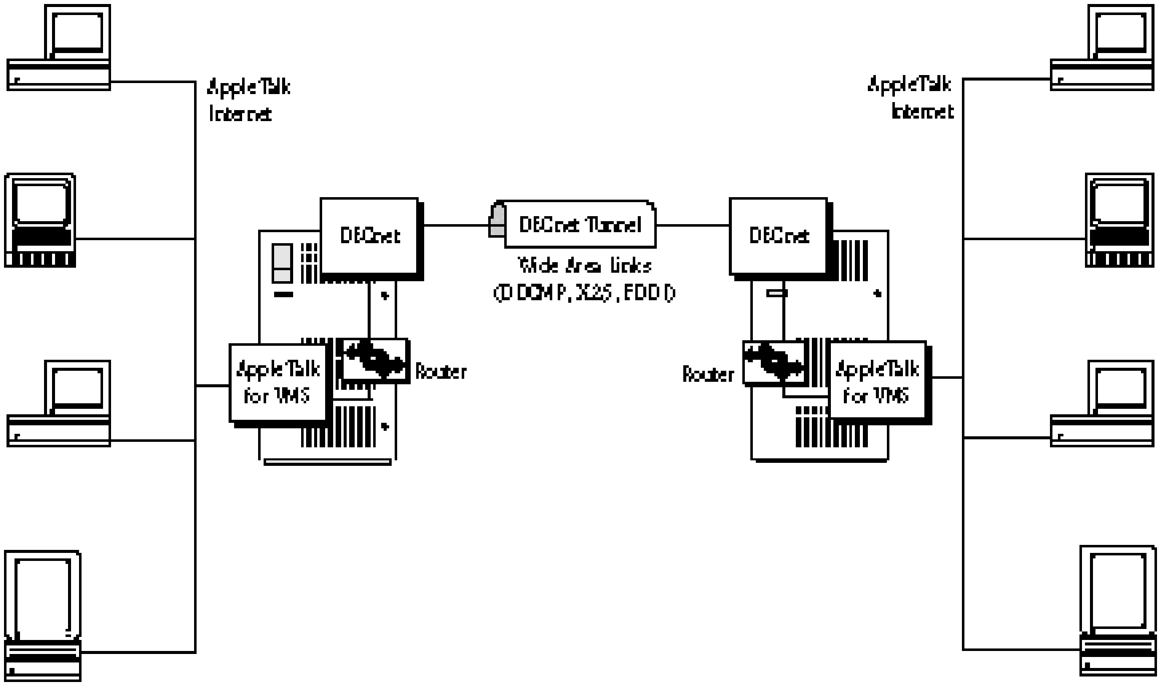


Figure 2-5 AppleTalk for VMS routing

How AppleTalk for VMS works

AppleTalk for VMS version 3.0 has four main parts:

- the AppleTalk Protocol Stack Driver
- the Ancillary Control Process
- the Protocol Interface Library
- the Configuration Program (ATK$MANAGER)

The AppleTalk Protocol Stack Driver is a standard VMS I/O driver that implements the main capabilities of the AppleTalk protocol suite. The Ancillary Control Process works with the Protocol Stack Driver to implement the more complex protocol functions, such as creating processes, authenticating users, and managing AppleTalk for VMS.

The Protocol Interface Library is the applications programming interface with the AppleTalk Protocol Stack Driver. The library is a set of simple subroutine calls, each implementing a different protocol function. Using these subroutines, applications can perform protocol functions such as looking up names, executing transactions, and transferring data. The Protocol Interface Library conforms to the VAX procedure-calling standard and can be called from any VMS system-supported programming language.

The Configuration Program is the user interface through which VMS system managers set up and observe AppleTalk for VMS. With this program, a system manager can perform such tasks as starting the AppleTalk for VMS router, opening and closing router ports, and reading performance counters.

DECnet

Just as AppleTalk interconnects Apple networks, Digital's DECnet/OSI networking software interconnects Digital enterprise networks. VMS, ULTRIX, MS-DOS, OS/2, and Macintosh systems can all be nodes in a DECnet/OSI network and communicate with each other. Figure 2-6 shows the DECnet protocols.

DECnet is the collective name for Digital's software and hardware communications products that implement the Digital Network Architecture (DNA). DNA is a set of protocols governing the format, control, and sequencing of message exchange for all Digital network implementations. The protocols are layered, and they define rules for data exchange from the physical-link level through the user-interface level. DNA controls all the data that travels throughout a Digital network. DNA also defines standard network management and network generator procedures. In most activities, however, these interfaces and protocols are transparent to the user.

How DECnet works

To communicate over a DECnet/OSI network, applications call DECnet services. Since DECnet handles all the network processing, applications running under different operating systems and written in different languages can exchange data over a DECnet logical link.

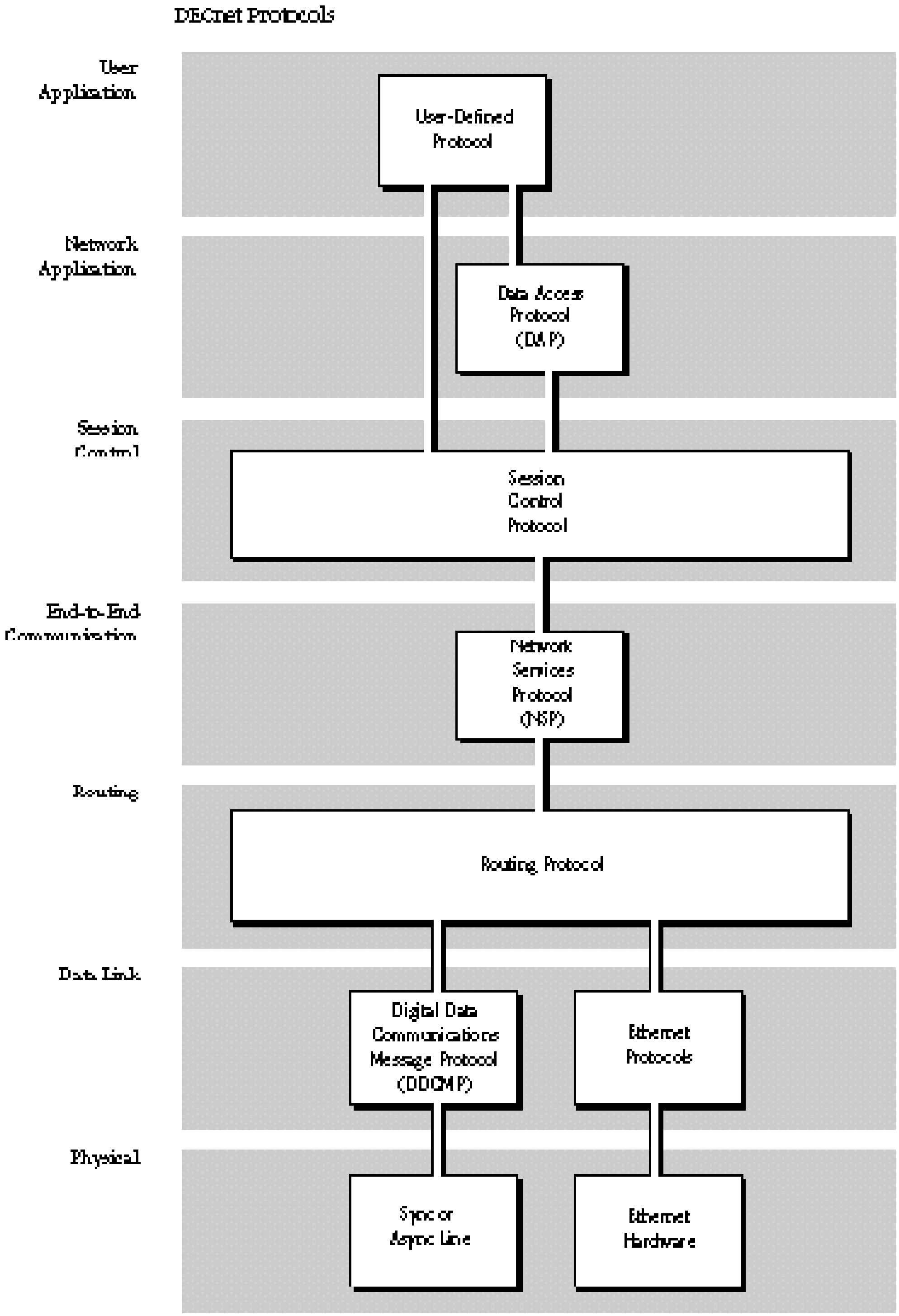


Figure 2-6 DECnet protocols

DECnet for the Macintosh allows a Macintosh computer to participate in any DECnet/OSI network. This means that direct connections can be made from a Macintosh to any DECnet application or service without having to be routed first to a VMSsystem-based VAX or other system. Because DECnet for the Macintosh includes a connection tool for the Macintosh Communications Toolbox, Macintosh applications written to the Communications Toolbox can work transparently with DECnet.

Benefits of DECnet

Through the DECnet/OSI network, Macintosh users and applications can

- directly participate in Digital's network management facilities
- start remote processes anywhere on the network
- copy files to and from Macintosh computers and other systems running DECnet
- build distributed applications that require wide area task-to-task communications with minimal overhead
- use wide area network terminal communications
- directly access Digital's SQL/Services™, VTX infobases, VAX Notes™ conferencing, and other DECnet-based services and applications

DECnet for Macintosh software works over a wide range of communications media, including Ethernet 802.3, LocalTalk, and asynchronous modems, so that users can connect to the network from their offices, from home, or from the road.

Digital plans to migrate DECnet toward the Open Systems Interconnect (OSI) protocols. Apple-Digital customers now have a number of migration alternatives to OSI: AppleTalk integration with DECnet, native DECnet on the Macintosh, and Apple or third-party OSI-compliant products on the Macintosh.

AppleTalk-to-DECnet

transport gateway

Although both AppleTalk and DECnet are based on layered architectures, the protocols differ between the systems. To link the two sets of protocols, Apple and Digital have developed the AppleTalk-to-DECnet transport gateway.

The AppleTalk-to-DECnet gateway is VMS-layered software that offers simple, reliable connections between AppleTalk and DECnet/OSI networks. The gateway is a bidirectional transport relay that provides end-to-end session links between clients or servers communicating with entities in the other network. Figure 2-7 illustrates the transport gateway linking two protocol stacks.

Through the gateway, Macintosh applications can use the AppleTalk Data Stream Protocol (ADSP) to communicate with applications that use Digital's Network Services Protocol (NSP).

The AppleTalk-to-DECnet gateway allows developers to write applications that easily communicate across Apple and Digital networks. Through the gateway, Macintosh applications gain access to DECnet-based services and applications.

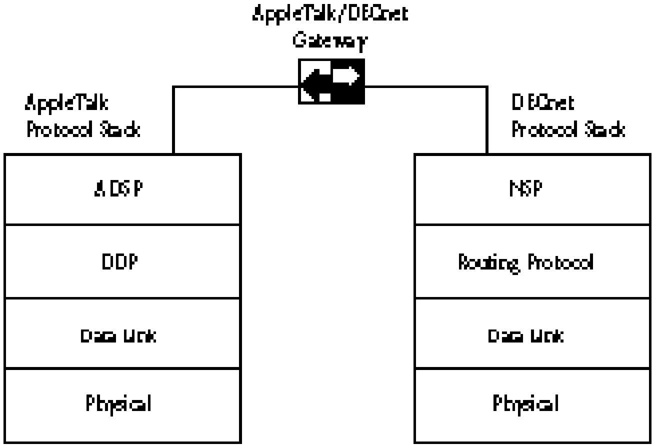


Figure 2-7 AppleTalk-to-DECnet transport gateway

To use the gateway, applications must incorporate access routines into their Macintosh or VAX clients or servers. On the Macintosh side, the Transport Gateway Access Tool of the Macintosh Communications Toolbox coordinates the connection setup and tear-down of the session. These access routines help the application connect to the gateway and relay the final destination to the gateway. The gateway then makes a DECnet connection to the final destination. Once the AppleTalk and DECnet sessions are running, the gateway relays data between them. If one side breaks the session, the gateway notifies the other session of the event.

From a Macintosh, the AppleTalk-to-DECnet gateway appears as any other service on the AppleTalk internet. A Chooser document is included with the AppleTalk-to-DECnet gateway to make selection easy for users. Indeed, the gateway registers its name on the internet in much the same way that AppleShare servers or LaserWriter printers register their names.

Network management

Shared applications depend on the stable environment that proper network management provides. This section outlines Apple and Digital strategy regarding Apple-Digital networks.

DECnet/OSI offers enterprise-wide network integration and management to Macintosh desktops. Therefore, Digital's comprehensive line of network management products covers the management of Macintosh computers that run DECnet. Currently, through Apple's Inter•Poll® network management software, network administrators can view system components of AppleTalk for VMS just as they view components of other AppleTalk internet members.

Among the capabilities in AppleTalk for VMS version 3.0 for controlling the network are opening and closing ports on the AppleTalk for VMS router, restarting the router and the transport gateway, and setting the maximum sessions allowed on the transport gateway. In addition, an administrator can manage the AppleTalk for VMS router from a remote Macintosh computer, using AppleTalk network management protocols, or from DECnet/OSI, using a configuration and control program. This program can be run either on the local VMS system or through a DECnet connection to a remote system.

From an Apple-Digital internet, network administrators will be able to

- view and manage Macintosh nodes with DECnet/OSI network management tools
- view remote AppleTalk networks through DECnet tunnels with AppleTalk network management tools
- monitor and control the components of AppleTalk for VMS and the AppleTalk-to-DECnet transport gateway

In the future, network management products from Apple and Digital will make even more extensive use of the management functions described here. For example, Digital's Enterprise Management Architecture will allow for remote management and control of enterprise networks from a single location.

3 Building Distributed Applications

A distributed application is one that shares information and is split—or distributed—among two or more computers. Apple and Digital are committed to providing tools that make it easy to build distributed applications for the Apple-Digital environment.

To build a distributed application, one of the first steps is to identify the functions that make up the application and decide how they should be shared among systems. The main techniques for sharing are

- distributed user interface
- distributed data access
- distributed computing

Often, more than one of these techniques is used in a single application. To choose the right technique for their applications, developers should analyze their requirements for performance, data sharing, and ease of programming.

Distributed user interface

The most common technique for distributing applications is the distributed user interface. With this technique, a developer can tailor the user interface of the host application so that it takes advantage of personal computer capabilities. For example, the developer can let the Macintosh computer manage the user interaction of a VMS application, thereby taking advantage of the powerful, easy-to-use Macintosh user interface.

A distributed-user-interface application consists of two parts:

- a personal computer (the front-end Macintosh)
- a host computer (the back-end VAX)

The Macintosh controls the user interface; the VAX controls the application logic and storage. For example, an application might store shared information on a VAX computer. The user interface, however, might run on a Macintosh personal computer, allowing the user interface to be customized for different environments.

Developing a remote user interface makes it easy to migrate from a centralized application that uses terminals to a distributed application that uses the power of personal computers on the desktop. Because a single VAX computer does the information processing, that computer easily coordinates shared access to information. In addition, because the user’s Macintosh controls the user interface, the application can be more responsive and easier to use.

Here are some current tools that help Apple-Digital developers provide remote user interfaces:

- terminal emulation
- the X Window System
- MacWorkStation™
- HyperCard®

Terminal emulation

Apple and third-party developers offer terminal emulation products for the Macintosh computer. MacTerminal® is Apple’s telecommunications/terminal emulation application program. This versatile program turns a Macintosh computer into an intelligent terminal that can communicate with a wide range of host computers including on-line information services, other personal computers, and Digital’s VAX and DECstations systems. MacTerminal (version 3.0) has been developed based on the Macintosh Communications Toolbox. This program supports multiple sessions, allowing the user to establish multiple host connections in different windows. MacTerminal offers many popular terminal emulation, file transfer, and connection capabilities, each implemented as communications tools:

Connection tools	AppleTalk ADSP Connection Tool
	Serial Connection Tool
	Apple Modem Connection Tool
	LAT Connection Tool
	DECnet CTerm Connection Tool
Terminal emulation tools	Teletype (TTY) Terminal Emulation Tool
	VT102™ Terminal Emulation Tool
	VT320™ Terminal Emulation Tool
File transfer tools	Text File Transfer Tool
	XMODEM File Transfer Tool

MacTerminal is completely modular; therefore, new tools developed by Apple or other commercial developers can be dropped into a user’s System Folder and become immediately available to the MacTerminal application. For example, MacTerminal can use the CTerm Connection Tool to provide terminal emulation capabilities in a wide area network.

How LAT works

Digital’s LAT (local area transport) communications protocol supports high-speed asynchronous communications for terminals and other devices connected through Ethernet local area networks (LAN). The LAT protocol provides an efficient connection between terminals and services on a network and minimizes the VAX computer processing resources required to provide the services. In cases where more than one VAX on a network offers identical services, LAT also provides “load balancing” by evenly distributing the demands for those services between the VAX computers offering them.

LAT implementation on the Macintosh includes a LAT driver, a control panel device (CDEV), and a connection tool for the Macintosh Communications Toolbox. LAT support on the Macintosh provides the following benefits:

- The protocol provides support for efficient terminal connection in Ethernet environments.
- The protocol provides multisession support over one Ethernet connection.
- Because LAT has been implemented as a connection tool for the Macintosh Communications Toolbox, any application built on top of the Communications Toolbox can make use of the LAT transport.

The X Window System

Apple and Digital support the industry-standard X Window System (X), version 11 (X11), developed at the Massachusetts Institute of Technology (MIT). X11 is a set of software components that allows developers to build applications with distributed, graphics-based, and hardware-independent user interfaces. All major desktop platforms can access applications written to the X11 standard, so that developers are ensured that their applications will be easily accessible to all environments.

Unlike most other windowing systems, X11 offers network transparency. That is, the user’s interaction with an application is consistent, whether the application is running locally or on another computer across the network.

MacX™, Apple’s X11-compliant server that can run in Digital’s DECwindows™ software environment, is included as part of the Apple-Digital agreement. DECwindows client applications can use the MacX server to implement a distributed user interface. Figure 3-1 shows how applications run on X11 clients and how users interact with applications on X11 servers.

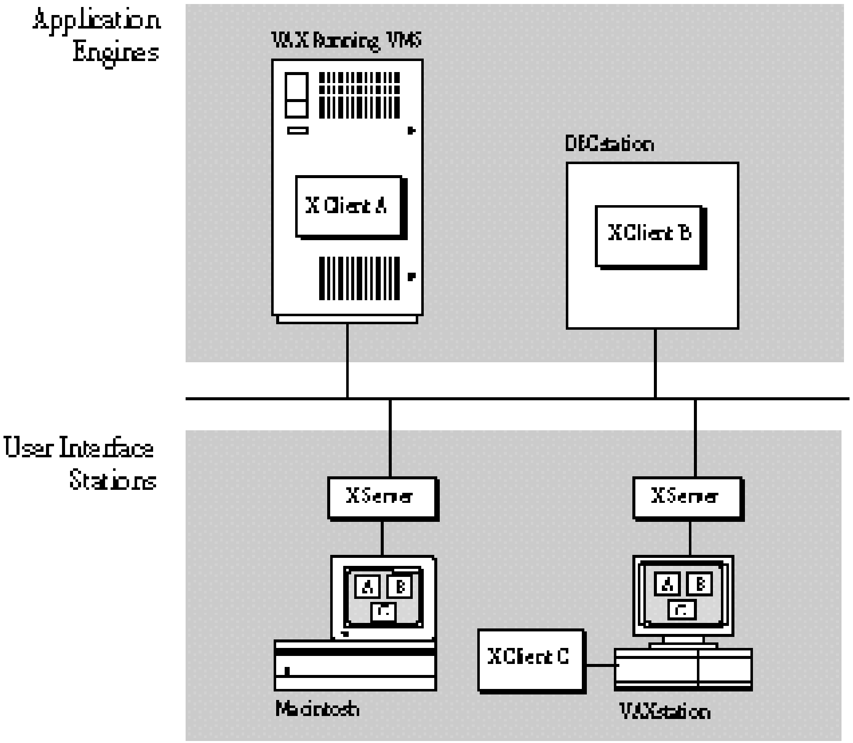


Figure 3-1 X Window System network transparency

Components of the X Window System

The X Window System has three main components: the X server, Xlib, and X Toolkit.

X server The X server runs on the user’s computer, providing low-level graphics, windowing, and user input functions. X clients com-municate with an X server through the network or “wire” protocol, also known as the X protocol. The X protocol can be exchanged between X clients and X server over different transport protocols such as the protocols of DECnet/OSI, TCP/IP, or AppleTalk.

Xlib Xlib is a library layered on top of the X protocol. It allows applications to access the protocol through a procedural interface. In addition, Xlib provides a set of utilities that most X applications need. The Xlib interface is the lowest level interface that applications should use; they should neither use the workstation hardware interface directly nor directly generate X protocol requests.

X Toolkit The X Toolkit, like Xlib, provides a set of utility routines for implementing user interfaces. The X Toolkit consists of two logical components, the intrinsics library and the widgets library. Widgets are routines that implement user interface features such as scroll bars, dialog boxes, and text fields that can be edited. Intrinsics are routines that make writing widgets easier.

Benefits of the X Window System

X Window System compliance benefits the user in several ways:

- DECwindows and Apple’s X server implementation are compatible with the industry-standard X Window System.
- Because all major desktop platforms can access applications written to the X Window System standard, developers are ensured that their applications will be easily accessible to all environments.
- Because the X Window System works in a distributed environment, a developer can run the heavy processing portions of an application on the VAX computer while using the Macintosh for optimal user interface functions.

DECwindows

Digital’s DECwindows software is an implementation and extension of X11. DECwindows runs under the VMS and ULTRIX operating systems, providing the Xlib and X Toolkit, as well as extensions to X11.

In Figure 3-2, the X protocol marks the division between X client and X server. On the VAX client side, the “staircase layering” of the application layer shows that applications can intermix calls to any of the client-side libraries. In addition to the standard X11 components, these libraries provide the following extensions:

- *DECwindows Toolkit.* DECwindows provides VMS and ULTRIX libraries for user interface primitives (widgets), resource management, and internationalization. Additional development tools are also included. With the DECwindows Toolkit, a developer can create features such as scroll bars, dialog boxes, text fields that can be edited, and menus.
- *Industry-standard interfaces.* In addition to the X11 interfaces, DECwindows offers industry-standard libraries such as PHIGS and GKS. Developers can use these interfaces when they want to minimize the cost of supporting multiple graphics environments, multiple windowing environments, or both.
- *Extension libraries.* X11 provides a mechanism for extending the server's capabilities. By taking advantage of this feature, the DECwindows architecture provides support for Display PostScript, 3D graphics, and imaging capabilities. Currently, these extensions are only available on VAX workstations.
- *Base applications.* DECwindows includes a set of useful base applications, such as window and session managers, terminal emulators, and personal productivity tools.

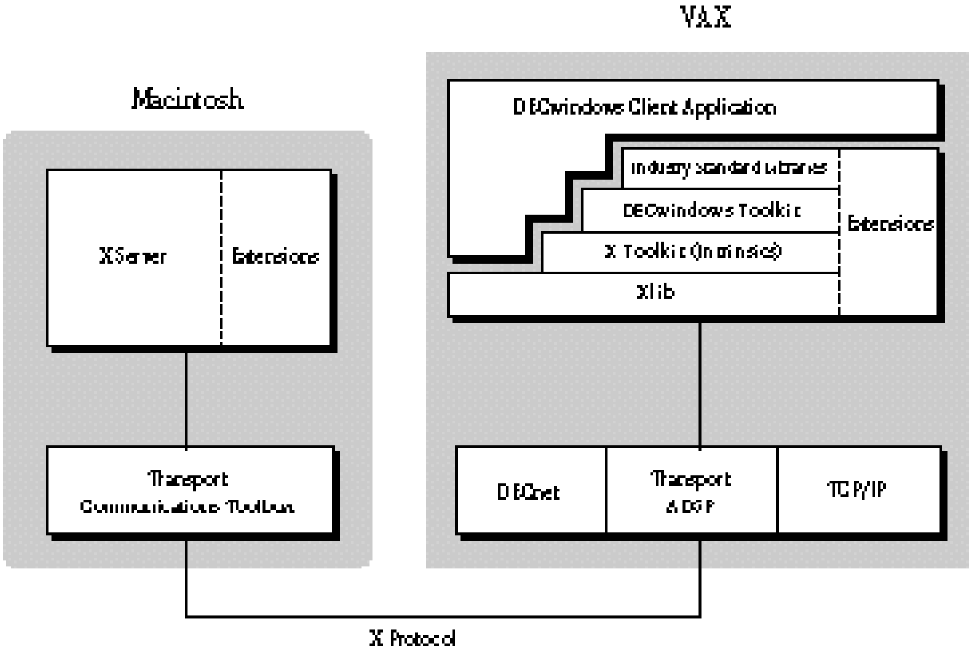


Figure 3-2 DECwindows and the Macintosh X server

DECwindows software enhances X11 by using faster window algorithms, exposing additional interfaces, allowing more flexibility, providing international keyboard support, and implementing a font-caching system. In addition, the DECwindows Xlib implementation checks for illegal parameter values, which greatly speeds application development.

In addition to language support found in X11, DECwindows supports the UNIX implementation of the C programming language and the complete set of VAX system- supported languages, including Fortran, ADA, and Pascal. To promote ease of use and user interface consistency among applications, DECwindows includes a set of guidelines, the *DECwindows User Interface Style Guide*.

The Macintosh X server

MacX, Apple's implementation of X11, consists of an X server running under the Macintosh Operating System. Through this server, a user can interact with X client applications running on remote hosts, such as VMS systems. The X server runs on the user's Macintosh, performing all graphic display operations and managing the mouse and the keyboard.

A host system running X clients can use the Macintosh X server for operations that involve the display, keyboard, or mouse without changes to the client programs.

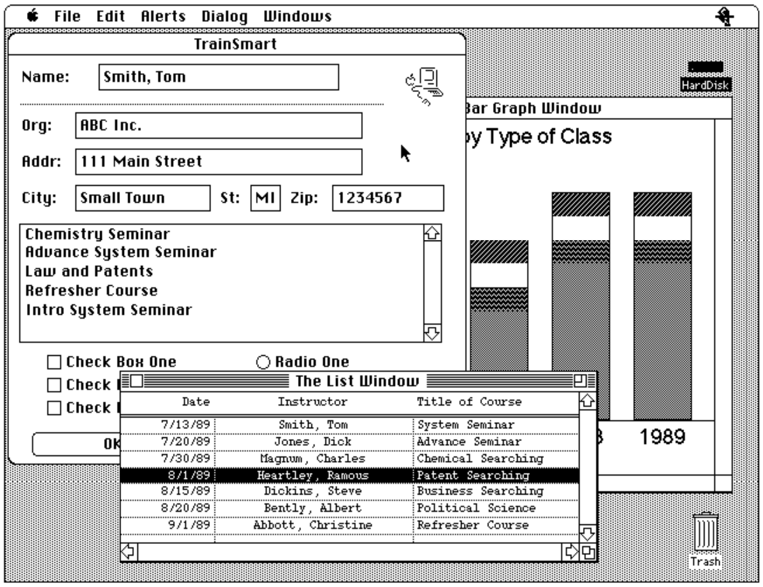
The following list presents key features of MacX:

- The X server supports monochrome and 8-bit color X client applications. It also supports multiple physical screens attached to one Macintosh computer.
- The Macintosh X server is compatible with MultiFinder®. In addition, text and graphics placed into the cut/paste buffer of the X server can be pasted into Macintosh applications by way of the Clipboard, and vice versa.
- The Macintosh X server supports concurrent connections to X clients by way of several network protocols through the Macintosh Communications Toolbox: AppleTalk (through the AppleTalk Data Stream Protocol), DECnet, or TCP/IP. A special interface is provided for the VMS system to allow DECwindows applications to run over AppleTalk.

- The X server includes its own window manager, thereby giving the Macintosh user the choice between two types of user interfaces: The interface can be either similar to the Macintosh (when the built-in window manager takes control of operations), or similar to DECwindows (when the built-in window manager is deactivated).
- The X server supports DECwindows Fonts.

MacWorkStation

MacWorkStation is a distributed user interface technology similar to X Windows. It is designed to leverage desktop resources for transaction processing applications. Its high-level message protocol and intelligent server application generate minimal over-head as compared with other windowing systems. Any host program over any speed network can use the MacWorkStation protocols to request user interface, printing, and filing services from the MacWorkStation server. Host applications can provide a graphical user interface while reducing host system processing and communications loads. Figure 3-3 shows an example of the MacWorkStation graphical user interface.



TrainSmart

Name: Smith, Tom

Org: ABC Inc.

Addr: 111 Main Street

City: Small Town St: MI Zip: 1234567

Chemistry Seminar
Advance System Seminar
Law and Patents
Refresher Course
Intro System Seminar

□ Check Box One
□ Check
□ Check

○ Radio One

The List Window

Date	Instructor	Title of Course
7/13/89	Smith, Tom	System Seminar
7/20/89	Jones, Dick	Advance Seminar
7/30/89	Magnum, Charles	Chemical Searching
8/1/89	Heartley, Ramous	Patent Searching
8/15/89	Dickins, Steve	Business Searching
8/20/89	Bently, Albert	Political Science
9/1/89	Abbott, Christine	Refresher Course

Figure 3-3 MacWorkStation’s graphical user interface

The MacWorkStation server locally manages the maintenance of all the standard objects without host processing or the need for the host programmer to learn how to program a graphical user interface. The host application can remain centrally controlled and updated, while distributing the heavy processing required to support a graphical user interface to the desktop.

With MWS Event Handler™ and MWS Dialog Builder, a developer can locally design and prototype the application user interface on the Macintosh without host or local code. The developer can then use the prototype as the production program by transferring the MWS commands up to the host program.

Figure 3-4 shows the Macintosh computer running the MacWorkStation server application and providing user interface services to the host program (called the client program).

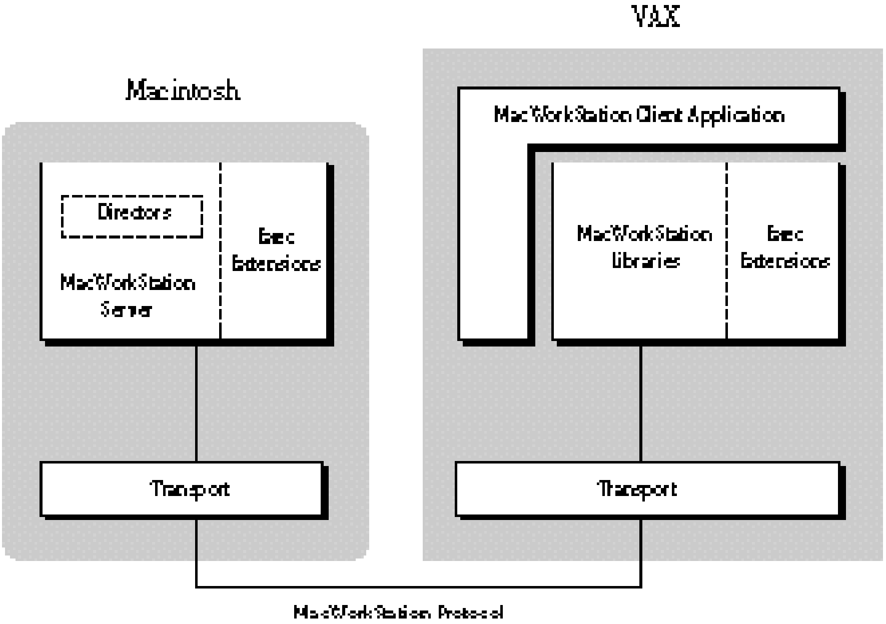


Figure 3-4 Macintosh computer running MacWorkStation server application

How MacWorkStation works

A programmer can write the host application in any language, but it should be structured in an event-driven manner to be easy to use and consistent with local applications. Third parties provide libraries to aid in the development of the host program. MWS protocols are sent to MacWorkStation and received in the same manner that text is printed and read from a terminal.

The MWS protocol is a high-level message system that is easy for host programmers to learn and use. The protocol runs over any cabling scheme or over any network: coax/twinax communications cards and LocalTalk, Ethernet, and token ring networks. Using the protocol, a programmer can create, update, and delete local objects from the host. The syntax is text based to ensure communications transparency across any network.

The MacWorkStation server is a Macintosh application that provides the user interface, printing, and filing services for the host client program. The server also has communications and scripting capabilities, as well as the ability to be extended with custom code.

Benefits of MacWorkStation

The distributed user interface technology that MacWorkStation offers affords users the following benefits:

- A developer can create a host application with a transparent Macintosh user interface.
- Utility programs allow applications to be designed quickly.
- Host processing is significantly reduced by distributing the creation, updating, and maintenance of user interface objects to the Macintosh computer. Graphic objects, such as menus and dialog boxes, can be stored locally without having to be re-created each time from the host.
- Code modules (called EXECS) can be used to expand MacWorkStation in a transparent manner and to distribute application logic from the host to achieve a cooperative application environment.

HyperCard

HyperCard is a personal software tool kit and information manager for the Macintosh computer. It offers unique ease of use, learning, and speed for building custom Macintosh software. Because HyperCard can be extended with code written in more traditional development environments, it is an ideal tool for creating custom user interfaces to virtually any type of media, including text, graphics, sound, animation, and video. Major software publishers provide tools to add connections to HyperCard for host connectivity, artificial intelligence, SQL database access, and multimedia device control.

How HyperCard works

HyperCard can be used in two different modes: browsing or scripting. The browsing mode lets users navigate through large quantities of information by means of the intuitive and easy-to-use hypertext metaphor.

The scripting mode offers capability for building customized front ends by using a sophisticated scripting language, similar to English, called HyperTalk®. This scripting language can be extended by adding sets of extensions to HyperCard. These extensions are called external commands (XCMD) and external functions (XFCN).

Apple provides separate sets of extensions (called tool kits) for different environments. For example, APDA offers tool kits to interface with AppleTalk, MacTCP®, and Data Access Language™.

Distributed data access and information sharing

In distributed data access, the application resides on the user’s personal computer, and accesses information stored on a host computer. The application achieves this record-level access by using tools for distributed database access, such as Apple’s Data Access Language or Digital’s SQL/Services product.

Using a VAX computer as a file server offers several advantages. For example, little or no change is required to existing Macintosh applications for them to take advantage of information and disk sharing. Yet the system provides increased file storage capacity, improved data consistency, and enhanced data security.

One of the many advantages of computing in a network environment is the ability to access and manipulate information from many resources. Apple and Digital provide a number of services that allow users to share, gather, store, and distribute information from various sources located on the network, without having to be concerned about the differences between VAX and Macintosh computers.

With these services, users can

- access and share files stored on computers on the network
- access and share printers connected to the network
- access and manipulate databases located on VAX computers on the network
- create documents in a standard format to facilitate interchange among users and applications on any computer on the network

File and print services

One of the most versatile and reliable information- and resource-sharing services available for communicating between VAX and Macintosh users is AppleShare,

which provides file services and print services. AppleShare’s file service is based on the AppleTalk Filing Protocol (AFP), and the print service is based on the Printer Access Protocol (PAP). Both AFP and PAP are unique features of the AppleTalk protocol stack.

Digital’s VAXshare™ file and print services are also based on AFP and PAP, respectively. Thus, the interface used to represent files stored on a file server or printers available on a network is consistently maintained, whether the services are on a VAX or some other hardware platform. In addition, VMS, Macintosh, MS-DOS, and Apple II users and applications can share the same folders and files.

VAXshare file service

VAXshare file service provides the following features:

- *Sharing of Macintosh files by using high-capacity VAX disks.* Because VAXshare file service is AppleShare compatible, a user can access both local and remote Macintosh files in the same way. The full range of file operations, such as reading, writing, renaming, and deleting files is available. The file service also provides file and byte-range locking for file sharing between systems, as well as access control and full Desktop/Finder™ integration.
- *Security for data.* VAXshare file service offers the user VMS file system security. AppleShare log-on sequences use the VMS user name and password; file access and privilege use VMS access and file protection.
- *File sharing between VMS and Macintosh users.* VMS and Macintosh users and applications can access the same files and VMS directories (volumes or folders). For example, Macintosh users can access VMS directories as Macintosh folders. In addition, Macintosh users and applications can use the VMS Distributed File Service to access files on other Digital systems connected to the file server.

VAXshare file service maps the Macintosh hierarchical file structure onto the VMS file structure by equating Macintosh volumes and folders with VMS directories and subdirectories, respectively, and Macintosh file data forks with VMS files. Each Macintosh file resource fork uses a separate VMS file.

Additional VMS files, hidden from the Macintosh user, contain other Finder and Desktop information needed to ensure AppleShare compatibility. The file service automatically and dynamically creates and maintains this additional information; thus, the Macintosh user can access VMS directories and files that were created by VMS users or applications.

The Macintosh user gains access to a VMS system-based file server by selecting an AppleShare server with the Chooser and logging in with a user name and password. The user name and password can be the same as the user’s VMS account.

The Macintosh user or application can create, access, and manipulate files just as the user would with any other AppleShare server. VAXshare file service and AppleTalk ensure that the

Macintosh file and Desktop environment is maintained. With VAXshare file service, Macintosh users can share graphics, natural image pictures, and even stereo sound. Macintosh applications that want to manipulate shared data can use file and byte-range locking to synchronize simultaneous data access in the same way as with AppleShare.

VMS users and applications can access Macintosh file data because the data fork of the Macintosh files is represented on the VAX system as a VMS stream file. The VMS user or application can create new directories or files. The first time a Macintosh accesses the directory or file, VAXshare file service automatically creates the necessary auxiliary files to implement the Macintosh view of the file.

VAXshare print service

VAXshare print service provides the following printing capabilities:

- Macintosh users can print files on Digital high-speed, high-resolution printers connected to a VAX computer. The VMS Distributed Queuing Service allows access to printers on other VAX nodes.
- Macintosh users have access to VMS queuing services to spool output for Apple's LaserWriter printers. A print spooler receives a print stream from AppleTalk, writes it to a VMS file, and submits a print request to the VAXshare print service.
- A common print symbiont is available that allows VMS users to print files on LaserWriter printers connected to the AppleTalk network.

VAXshare print service supports the following printers:

- Digital PostScript printers on a VAX computer
  - PrintServer™ 40 Plus (LPS40 Plus)
  - PrintServer 20 (LPS20)
  - ScriptPrinter™ (LN03R™)
- Apple's family of LaserWriter printers on AppleTalk

Macintosh users and applications select and write to VAX printers in the same way that they select LaserWriter printers connected to an AppleTalk network. Digital printers connected to a VAX system appear on the Macintosh Chooser menu as additional printers on the AppleTalk network.

VMS, MS-DOS, and OS/2 users and applications access LaserWriter print queues in the same way that they access other VMS print queues, by using VMS system services utilities and the DCL PRINT command, or Digital’s PCSA™ software.

Remote database access

Desktop access to enterprise data is a significant issue for many companies today. Apple’s Data Access Language (DAL) and Digital’s SQL/Services software let remote Macintosh applications access VMS-based databases. Both DAL and SQL/Services

- use client/server technology
- can access information in VAX Rdb/VMS™ databases
- allow easy data definition and manipulation of those databases
- have a callable applications programming interface (API) on Macintosh and VMS systems, and on other major hardware platforms.

Data Access Language

Apple’s Data Access Language is a connectivity language that gives Macintosh applications access to data stored in relational databases on VAX or IBM systems. In addition to Digital’s Rdb/VMS database, DAL supports several other important data sources including Oracle, Ingres, Informix/SQL, and SYBASE database systems. It also supports DB2 and SQL/DS on IBM systems.

With DAL, Macintosh applications can

- determine what databases are available on a VMS host
- determine the structure on a VMS database
- query databases to retrieve data from the VMS system
- insert, delete, and update data in the VMS database
- use database facilities for transaction commit and rollback functions

Data Access Language insulates Macintosh client applications from variations in networks, in host operating systems, and in host database management systems. Figure 3-5 illustrates how DAL works.

Data Access Language is based on the SQL and C languages, with extensions both for access to nonrelational data sources and for network support. It features a uniform interface to host data, using standard DAL data types, SQL syntax, and error codes. DAL is available to Macintosh application developers through the Data Access Language Applications Programming Interface (API) on the Macintosh.

The network connection between the Macintosh and VMS systems can be a direct or dial-up serial connection or an AppleTalk network connection. For the AppleTalk connection, DAL uses the AppleTalk Data Stream Protocol and requires that AppleTalk for VMS be installed on the VMS host.

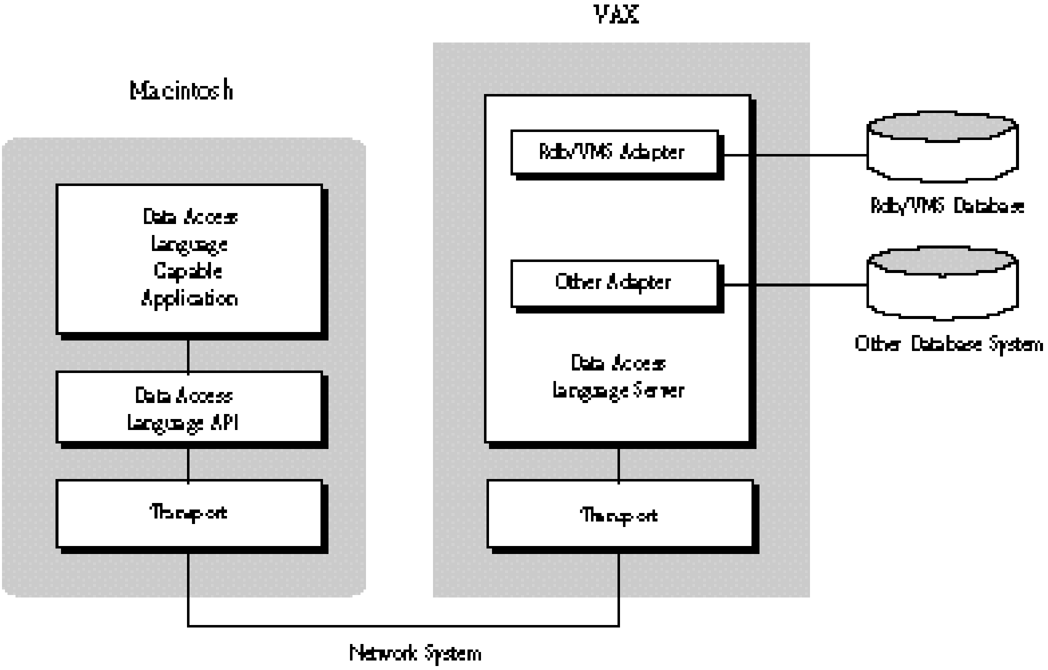


Figure 3-5 Overview of Data Access Language

How Data Access Language works

To use Data Access Language, a Macintosh client application sends requests for host data in the form of a DAL program. To send these requests to DAL and to retrieve the results, the client calls the Data Access Language API.

For applications that support external code (such as HyperCard's XCMDs), developers have found that a convenient way to support DAL with minimal effort is to package a set of external procedures that provide access to DAL from within the application's scripting or macro language. A set of XCMDs for HyperCard has been developed and is included in the Data Access Language Developer's Toolkit for the Macintosh as an example of how DAL can be supported. Also included in the Toolkit are programming interfaces for C and Pascal developers. Many commercial Macintosh applications have already taken advantage of the Toolkit and are supporting DAL. This support includes spreadsheets, databases, applications generators, expert systems, query tools, desktop mapping software, and other applications.

Data Access Language on the VMS system

The Data Access Language software on the VMS system operates as a server, accepting requests from a Macintosh client, carrying out those requests, and returning the results to the client for processing. The DAL server is a user-level VMS application that runs under standard VMS and database security, and is transparent to the client application on the Macintosh. There is no direct interface to the server on the VMS system. The DAL server is installed using the standard VMS installation utility.

Data Access Language on the Macintosh

A Macintosh application uses Data Access Language through the Data Access Language API presented on the Macintosh. The API is a collection of 12 function calls that allow the application to send a DAL program to DAL for execution, determine execution status, and retrieve results of the DAL request. All communication between the application and the Data Access Language API takes place using standard Macintosh data types.

The Data Access Language API is currently implemented on the Macintosh as a small collection of callable routines to be linked with the client application and a

DAL device driver, which is called by those routines. The Data Access Language API remains uniform, whether it talks to the DAL Server for VMS or to other DAL servers (for example, MVS, VM, and UNIX).

Benefits of Data Access Language

With Data Access Language, Macintosh developers can focus on integrating host data into the Macintosh application, instead of on networking and VMS programming. Macintosh applications can reach beyond local resources to access and manipulate shared data on the VMS system. DAL lets the Macintosh developer implement a single host data access facility, instead of separate facilities for each VMS database and network connection.

SQL/Services for VAX Rdb/VMS

Digital’s SQL/Services software extends the power and capacity of Digital’s database server environment to the desktop by incorporating existing and emerging industry-standard interfaces into Digital’s relational database platform. SQL/Services software lets remote desktop applications access VAX Rdb/VMS relational databases. The SQL/Services API for Macintosh access is consistent with the SQL/Services callable API offered by Digital on other platforms.

How SQL/Services works

SQL/Services client applications incorporating the SQL/Services API communicate by way of DECnet with a server process on the VAX system on which the target database resides. Users can select direct DECnet communication using the DECnet Tool in the Macintosh Communications Toolbox or through the AppleTalk-to-DECnet gateway. On the VMS system, SQL/Services uses the dynamic interface of SQL to access Rdb/VMS databases. Once an application is built with SQL/Services, all the user needs to access the target database is the Rdb/VMS run-time option, which is licensed with the VMS operating system. Figure 3-6 demonstrates remote data access using SQL/Services.

Applications developed using SQL/Services can also retrieve information from certain IBM-based databases, such as DB2, accessible through Digital's VIDA (VAX-IBM Data Access) interoperability software products. Applications can use the data accessed from an IBM mainframe as they would use data accessed from an Rdb/VMS database.

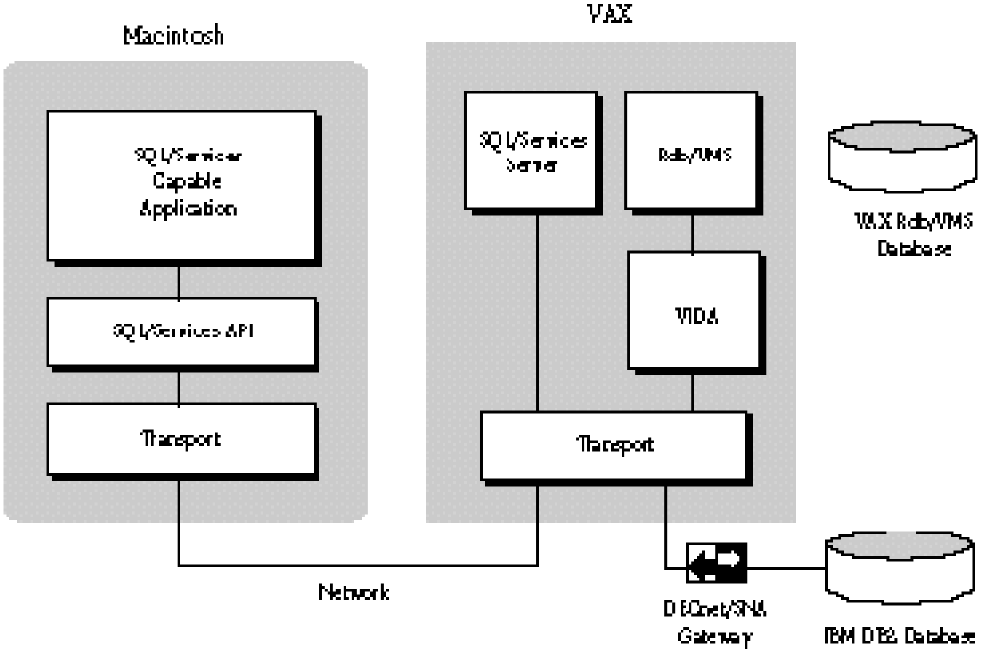


Figure 3-6 Remote database access using SQL/Services

Benefits of SQL/Services

Writing applications that use SQL/Services offers several benefits:

- SQL/Services is built upon ANSI-ISO standards. The Rdb/VMS SQL interface is Digital's implementation of the ANSI/ISO-adopted relational database language. By using SQL/Services, a developer can take advantage of SQL to gain access to Digital's database environment.

- SQL/Services gives direct access to the large installed base of established Rdb/VMS databases. Such access opens new markets for developers' applications.
- Digital also provides the SQL/Services API on VMS, ULTRIX, MS-DOS, and OS/2 operating systems. An application can depend on the same SQL calls being available on each of these systems, making it easier to port applications.

Compound document architecture

One of an organization's most persistent needs is to readily share compound documents—those that contain text, graphics, and data—among different applications and systems. Digital's CDA™ (compound document architecture) allows developers to write applications that manage compound documents easily, regardless of the environment in which the developers or application users are working.

This CDA architecture becomes especially useful now that heterogeneous computers that run applications with different file and document formats are connected to a network. Users need not be concerned with how a document that contains such data types as structured multifont text, spreadsheets, graphics, images, and application data is created and processed, or how other users will access the document.

CDA lets users treat all document-based information as flexibly as if it were simple ASCII text. CDA features robust interchange standards, which are consistent with the direction of both existing and emerging international and industry standards.

Users have two choices for using CDA in an application:

- read/write documents directly in native CDA format
- convert between the application's format and CDA

CDA is hardware and operating system independent, and provides the following set of standard data formats:

- DDIF™ is Digital's document interchange format for encoding revisable-form text, graphics, and image data on VMS systems. While maintaining a kinship with the Office Document Architecture (ODA) and other standards, DDIF also extends the capabilities of these existing standards to reflect the growing needs of document processing.
- DTIF is Digital's table interchange format used by VMS applications for the storage and interchange of documents that contain data tables, formulas, and spreadsheets. DTIF tables can be stored and referenced in DDIF-encoded compound documents.

Accessing CDA

On a VMS system, users access CDA services in the following ways:

- through the CDA Toolkit, a library of callable routines that enable developers to easily read and write compound document files
- through the CDA Converter, a callable service that lets developers "plug in" their own document format modules used in translating to and from DDIF
- using viewers, callable services that display formatted output data in a workstation window or character cell terminal
- using the MAIL utility, which lets developers exchange DDIF and DTIF documents

Document interchange

In an Apple-Digital internet, there is a need to exchange files in a format that will be recognized by both Macintosh and VMS applications. Apple and Digital have selected DDIF as one format for this exchange. Apple and Digital offer a set of converters running on the VAX to convert documents between Macintosh data representations and DDIF. Figure 3-7 illustrates document interchange using DDIF.

In some Macintosh files, certain application-defined formats are recognized by other applications to provide file compatibility—the major examples are the PICT file format and the formats used in MacWrite® and MacPaint®. Apple and Digital provide a set of converters that convert DDIF files to and from PICT, MacWrite, and MacPaint formats. Major software vendors provide additional converter modules to manage their application-specific file formats.

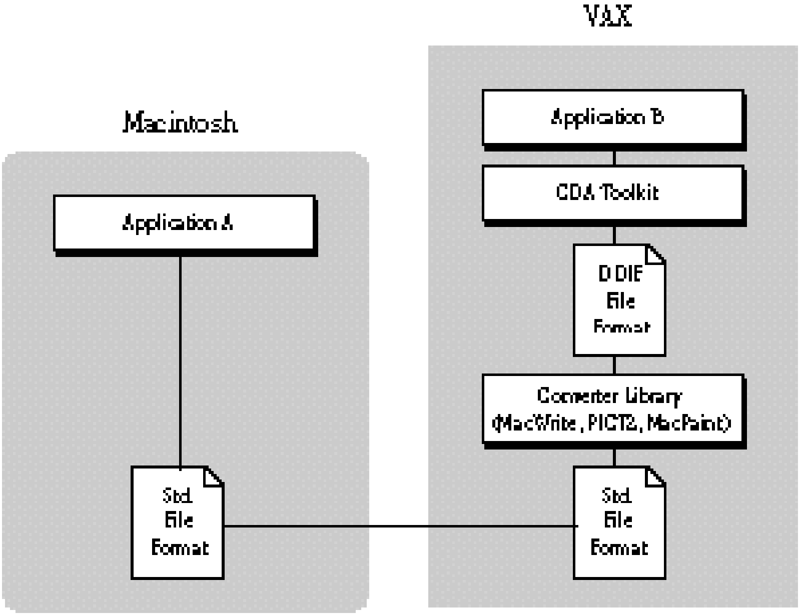


Figure 3-7 Document exchange using DDIF

Benefits of CDA

CDA provides users with several benefits:

- Applications are able to access the same revisable-form data.
- Functions such as file display and copying use standard facilities, thereby reducing the amount of code a developer must write.
- Data exchange across a network is in a format that can be recognized and processed by Macintosh or VMS applications. For example, a Macintosh user can mail or exchange a document with a VMS user or application, which can readily process that document.

Distributed computing

Distributed computing (processing) is the technique of sharing tasks among two or more computers. Many applications can benefit from this technique. For example, in such an application, processing that uses stored data could take place on the computer that stores the data; meanwhile, processing that involves complex user interaction could take place on the computer that provides the user interface. When designing a distributed Macintosh-to-VAX application, the following steps should be considered:

1. Decide how you want to divide tasks between the Macintosh and VMS systems.
2. Develop an application protocol to communicate application-level information between the two systems.
3. Choose one of the transport protocols offered by the Macintosh Communications Toolbox.
4. Continue to develop your application, using VMS software tools for the VAX portion and either the Macintosh Programmer’s Workshop (MPW®) or MacApp® for the Macintosh portion.

To enable developers to create applications in a distributed computing environment, protocols that allow this type of communication must exist. Within the AppleTalk and DECnet/OSI network systems, those protocols are the AppleTalk Data Stream Protocol and DECnet Task-to-Task. In addition, interprocess communication mechanisms and remote procedure calling expand the capabilities of distributed computing.

AppleTalk Data Stream Protocol

The AppleTalk Data Stream Protocol (ADSP) is the transport mechanism that underlies interprocess communication between Macintosh and VAX computers. ADSP is a symmetric, connection-oriented protocol that allows the establishment and maintenance of full-duplex data streams between pairs of sockets in an AppleTalk internet. ADSP provides flow control and the guaranteed, ordered delivery of data bytes.

Using ADSP, a client application can open a connection with a remote end, send data to and receive data from the remote end, and close the connection. The client can either send a continuous stream of data or break the data into client-intelligible messages. Additionally, ADSP provides an attention message mechanism that the client can use for its own internal control. A forward reset mechanism allows the client to abort the delivery of an outstanding stream of bytes to the remote client.

The features of ADSP closely match the needs of many distributed applications. ADSP offers a simple open/close/read/write interface and is symmetrical with NSP, the corresponding protocol of the Digital Network Architecture. Apple and Digital support ADSP as a transport mechanism for developing Macintosh-to-VAX applica-tions and for accessing the AppleTalk-to-DECnet transport gateway. Macintosh developers can choose to access ADSP by way of the ADSP connection tool from the Macintosh Communications Toolbox or by way of direct access to the driver.

Although ADSP is the preferred choice for developers, other AppleTalk protocols can be used to build Macintosh-to-VAX applications. These protocols include the AppleTalk Transaction Protocol (ATP), the AppleTalk Session Protocol (ASP), and the Printer Access Protocol (PAP). These protocols are supported both on Macintosh personal computers and on Digital's VMS systems. Note, however, that the transport gateway supports communication only through ADSP.

DECnet Task-to-Task

The DECnet Session Control layer can also be used when developing task-to-task applications between Macintosh and VAX computers. This layer defines the system-dependent aspects of process-to-process communications. The functions provided at this layer include name-to-address translation, process addressing, and access control.

The Session Control layer interfaces with the End-to-End Communication layer. NSP is the protocol used between the two End-to-End Communication modules. This layer controls the system-independent aspects of communications. These include connection management, data flow control, end-to-end error control, and segmentation/reassembly of user messages.

Programmers include DECnet calls in cooperating programs to allow them to communicate with each other. These calls activate routines that request the End-to-End Communications modules in each node to perform specific functions, such as the creation and control of a logical link.

DECnet Task-to-Task calls in communicating programs perform several functions, including requesting a logical link, receiving a logical-link request, and accepting or rejecting a logical-link request. In addition, task-to-task calls send data, receive data, send interrupt data, and receive interrupt data. A task-to-task call can also terminate a logical link.

In most DECnet implementations, performing task-to-task communication is similar to performing input/output operations. The logical link between the two communicating programs or tasks is like an I/O channel over which both programs can send and receive data.

Interprocess communication and
remote procedure calls

The goal of a well-designed distributed processing architecture is to complete processing quickly and with a minimum of communication among computers. Apple and Digital intend to offer two ways to distribute processing: connection-oriented interprocess communication (IPC) and remote procedure calls (RPC).

Interprocess communication, as its name suggests, is a technique by which computer processes communicate. Connection-oriented interprocess communication uses network connections to let processes on different computers communicate. When a developer designs an application that uses IPC, he or she designs a protocol—that is, a set of rules—for communicating information among parts of the application. The developer can then tailor the protocol to the application, to make distributed processing more efficient.

A remote procedure is one that can be executed from a network node other than the local node (the one running the main program). A remote procedure call (RPC) is a technique that makes calling a remote procedure as easy as calling a local procedure. RPC can make distributing an application easier, providing easier programming and debugging with no significant decrease in performance. In addition, a programmer can implement RPC in many existing applications without extensive reprogramming.

In the main program, calls to remote procedures look like calls to local procedures. The difference is that remote procedures are not linked with the main program and need not reside on the local node. Figure 3-8 shows how RPCs work.

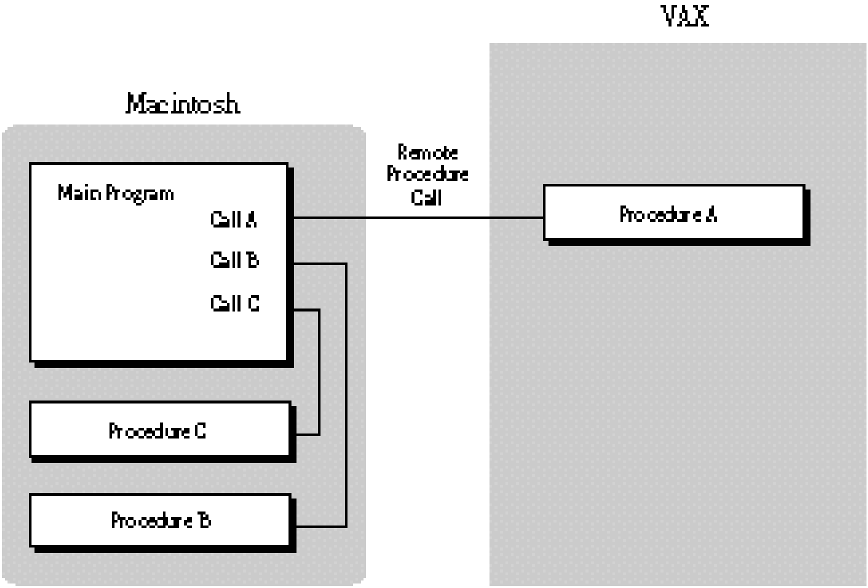


Figure 3-8 Conceptual model of remote procedure calls

In a remote procedure call, the application seems to call the remote procedure directly; in reality, the application calls the remote procedure processor. The remote procedure processor locates the server, connects it, then prepares the arguments of the call for transmission to the node that stores the remote procedure.

Any procedure can be made a remote procedure if it does not share global storage with other procedures. The calling program and the remote procedure do not have to be written in the same programming language, nor do they have to run under the same operating system. When the languages or operating systems differ, the RPC facility converts between the two data formats. Most RPC facilities can transparently locate the program being called by using a directory service.

Macintosh-to-VAX RPC integration is included in the design objectives of the Apple-Digital agreement.

An important part of the Apple-Digital environment is the communications services that let applications efficiently share and distribute information throughout organizations. Among these services are electronic messaging and mail, videotex services, and computer conferencing.

Electronic messaging and mail

Electronic messaging is a technique by which the computer systems of a network can exchange information reliably, though not in real time. The most widespread use of electronic messaging is interpersonal messaging, commonly called electronic mail. Other uses, such as Electronic Data Interchange (EDI), are becoming increasingly popular.

Digital offers a comprehensive international standards-based, messaging architecture and a range of products. These products include ALL-IN-1™ MAIL, the ALL-IN-1 Integrated Office System for work groups and departments, and the MAILbus™ family of products, which provide enterprise-wide messaging connectivity. Apple is committed to supporting message-based applications as an integral part of the Macintosh computer and the AppleTalk network system. Apple and Digital are committed to delivering direct access from Macintosh computers to Digital’s messaging services and to support the CCITT X.400 recommendations for message handling systems.

Applications for messaging and mail services

Using messaging and mail services, a developer can create many kinds of applications. For example, the developer can write applications that let users send and receive mail from any system on the network; one such Macintosh-based application is the ALL-IN-1 MAIL for Macintosh product. The developer can also write applications that perform other functions in addition to sending mail. For example, an application that sends reminders to insurance agents on their clients’ birthdays could also let agents create and edit reports, forms, and even customized birthday cards. A spreadsheet application might let users mail a spreadsheet to co-workers without leaving the application.

To understand the benefits of messaging requires first an understanding of the concepts behind the X.400 recommendations. These concepts define how systems send messages to one another and are important to the exchange of messages in a worldwide, multivendor network mail system.

The sections that follow offer a preview of opportunities for messaging and mail applications. In addition, these sections introduce the concepts behind the X.400 recommendations.

X.400 messaging

The X.400 recommendations are a set of protocols and guidelines developed by the International Telephone & Telegraph Consultative Committee (CCITT); these recommendations govern how messages are transmitted between systems. If systems (including multivendor systems) follow the X.400 recommendations and share a network, they can exchange messages with one another. Two software components make this exchange possible: the message transfer agent and the user agent.

- The message transfer agent controls delivery of messages throughout the network, providing guaranteed, store-and-forward transfer of messages to users on
  other systems.
- The user agent (UA) represents a single entity. It accepts messages on its user’s behalf, keeps track of the user’s mailboxes, presents messages to the user, allows messages to be created, and interfaces with the message transfer system on the user’s behalf.

The X.400 recommendations define protocols by which the various components communicate with one another. The two most important of these protocols are P1 and P2. The P1 protocol defines the interaction between message transfer agents. The P2 protocol defines the structure of X.400-compatible messages and defines the functional interaction between user agents of common service. Figure 4-1 shows how the X.400 protocols work.

Through Apple-Digital messaging and mail service, applications can conform to the P2 protocol and

- exchange mail with other Apple and Digital user agents
- interconnect between Digital and other mail systems by way of X.400 or proprietary gateways
- offer user features that extend beyond the X.400 recommendations

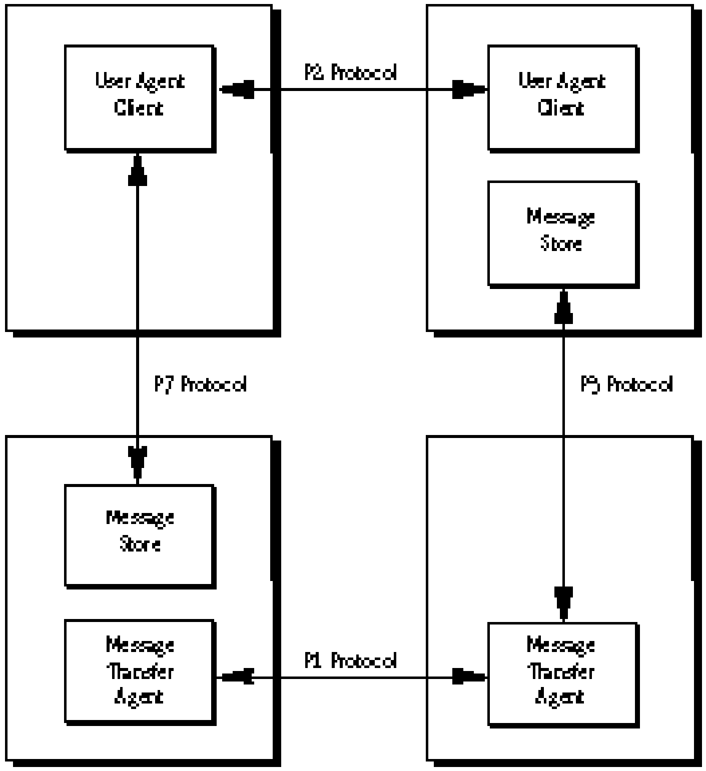


Figure 4-1 Generic model of the X.400 message handling system

ALL-IN-1 MAIL for Macintosh

The ALL-IN-1 MAIL for Macintosh application provides key electronic mail capabilities such as store and forward services and message-delivery notifications, as well as data about the message and the user. In addition, ALL-IN-1 MAIL for Macintosh implements all the mandatory services defined by the 1984 CCITT X.400 “P2” user agent recommendations. It also features

- access to Digital’s Distributed Directory Service (DDS)
- CDA architecture support for compound documents
- access to the full range of MAILbus multivendor interconnect services
- network filing services

ALL-IN-1 MAIL is based on the client/server implementation model. The ALL-IN-1 MAIL for Macintosh client application, residing on a Macintosh personal computer, provides the local services required by users to manage their personal mail system by means of a graphical user interface.

The services provided allow users to create, edit, file, and manipulate messages within their file cabinets. A personal address book is included for storing the name, nickname, phone number, and mail address for personal use. Users can access Digital's Distributed Directory Service for assistance in addressing other users anywhere on the MAILbus electronic mail network.

The ALL-IN-1 MAIL system supports delivery notifications, as well as indicators flagging the priority, importance, expiration date, and reply-by date. Users manage drawers and folders, either locally or on the mail server, where messages are filed. The Macintosh client can make a connection to the ALL-IN-1 MAIL Server for VMS either through the AppleTalk-to-DECnet gateway or directly to DECnet through the DECnet Tool in the Macintosh Communications Toolbox.

The ALL-IN-1 MAIL Server provides electronic mail services on behalf of all users within the local area network and remote users connected by way of asynchronous DECnet. The ALL-IN-1 MAIL Server receives messages and stores them in the users' server file cabinets. Users then access their mail when it is most appropriate for them. The ALL-IN-1 MAIL Server provides local message delivery services within a local area network and submits messages to Digital's MAILbus for remote message delivery. Messages can be exchanged with Digital's ALL-IN-1 Integrated Office System, other ALL-IN-1 MAIL servers, VMS Mail, as well as users of IBM PROFS, IBM DISOSS, and public and private systems conforming to the X.400 recommendations.

Figure 4-2 shows how ALL-IN-1 MAIL user agents fit into Digital's MAILbus Message Handling System.

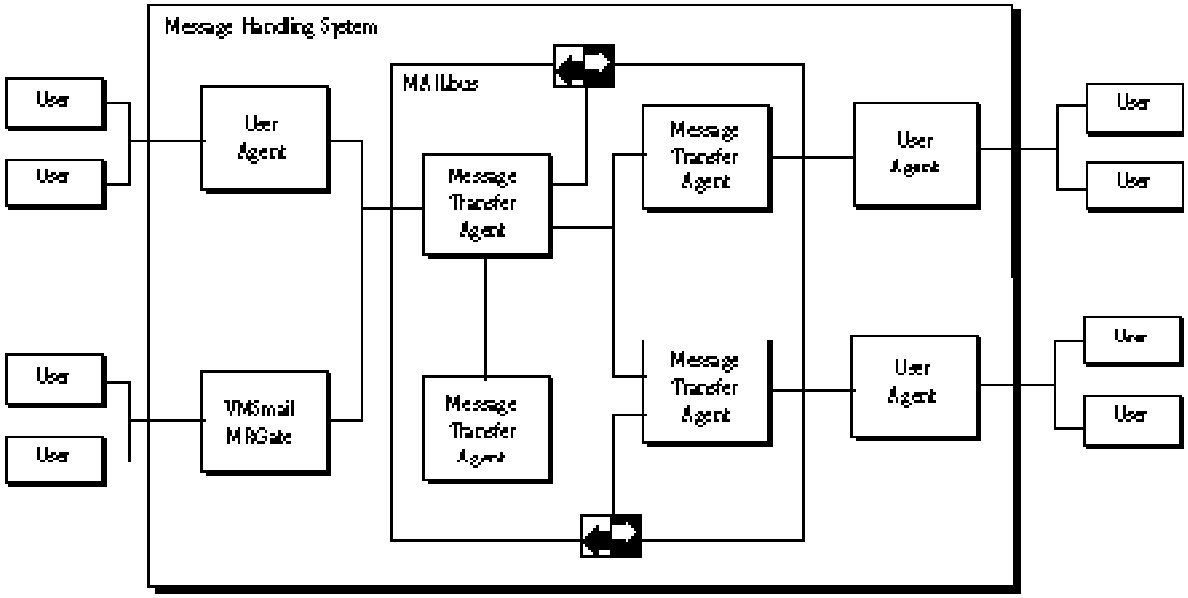


Figure 4-2 Digital’s message handling system implementation

VMS Mail access

Digital’s VMS Mail System is a widely used electronic mail utility included with the VMS operating system. Users can access the VMS Mail server from desktop computers, such as Macintosh or MS-DOS computers. This server will perform on the user’s behalf all the classical operations that VMS Mail allows, such as sending or reading mail, replying, forwarding, and so on.

A Macintosh client application, based on the Macintosh Communications Toolbox, can access the VMS Mail server and allow users to access their VMS mailbox from the Macintosh. This client application also permits local editing of messages.

Videotex services

With Videotex, a developer can create information applications, provide network access to centralized or distributed information, and build external applications through basic tools and programming interfaces. Examples of information that a developer can organize, maintain, and deliver as videotex information applications include policies and procedures manuals, sales jobs books, training schedules and course descriptions, CAD/CAM drawings, and newswire stories. VAX VTX™ is Digital's videotex product that retrieves and displays information.

The first implementation stages of the Apple-Digital agreement do not address videotex services, except to allow users access to VAX VTX through terminal emulation. It is Digital and Apple's strategy to offer Macintosh client software for VAX VTX in the future.

How VAX VTX services work

With VAX VTX, developers can build information systems that are varied and easily expanded. For example, a developer can first design a simple information retrieval or transaction-based system and then evolve it into a more complex system with the VAX VTX open distributed architecture. The open architecture lets users easily increase their systems' capabilities by adding new applications and servers.

Building distributed applications with VAX VTX services requires little or no knowledge of the underlying network. Currently, fast information access in a distributed environment is achieved by making VAX VTX available over networks with the DECnet/OSI architecture. Application users can retrieve data stored in any format, from any point on the network.

The components of VAX VTX are as follows:

- Infobase Server, a server that stores a database of formatted videotex pages
- Terminal Control Program/Concentrator (TCP/CON), which displays and retrieves an infobase page on a workstation
- External systems, a VAX computer or other host system running an application
- Update Server, a server that formats and loads information into an infobase

Figure 4-3 illustrates the VAX VTX architecture.

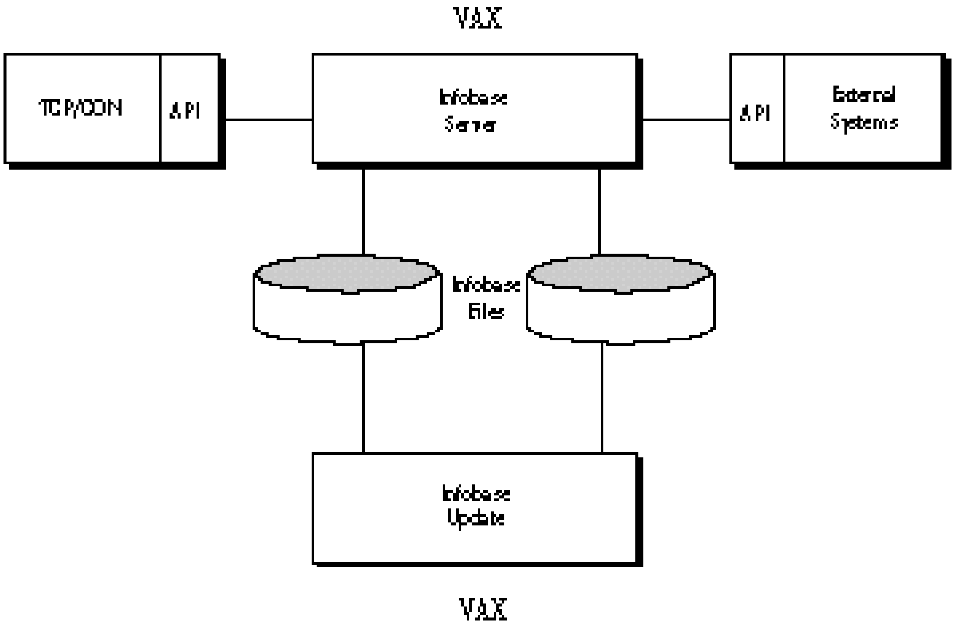


Figure 4-3 VAX VTX architecture overview

Benefits of VAX VTX services

VAX VTX services offer the following benefits:

- Because VAX VTX is not a turnkey system, most customers spend considerable resources defining and creating customized information access systems for their environments. This is a ready market for application development. Sample applications include telemarketing, newswire feeds such as AP NEWS, Dow Jones Service, inventory systems, and literature-ordering systems.
- On top of the existing open architecture, developers can create generic packaged systems or customized information solutions for customers, using both the front-end and back-end capabilities.
- Developers can access database-retrieval services to include many types of information under a single user interface. Information can be accessed through wirefeeds or from third-party services and loaded interactively or in batch mode, whether or not the service is actively running.

- Developers can build specialized free text search systems by adding a currently integrated solution or one personally designed. For example, a developer can add actual information such as an encyclopedia to this system and package it as a total service.

Computer conferencing

Computer conferencing is a technique that allows ongoing discussions among many computer users. At any time, users can write messages to conferences or read messages written by other users. Computer conferencing organizes and records discussions, making teamwork easier and reducing the need for conventional meetings. Among the many uses of computer conferencing are

- writing and reviewing documents jointly
- solving problems jointly
- brainstorming engineering designs
- consulting experts without interrupting them
- teaching and training
- managing projects

Digital’s VAX Notes conferencing software lets a user meet with other users electronically. With VAX Notes, users anywhere on a network can join ongoing conferences at any time. Conferences can reside anywhere on the network and can be accessed simultaneously by many users and systems.

In a VAX Notes conference, users organize their discussions into a series of topics and replies. By assigning keywords to a topic or a reply, a user can help others browse the conference. After a user reads a topic or reply, VAX Notes identifies it as “seen” so that he or she can then choose to read only unseen topics and replies.

Generally, a VAX Notes conference is managed by a moderator, who keeps the discussion on track. In addition, the moderator can open a conference to all users or restrict it to certain users, regardless of where the users reside on the network.

VAX Notes offers callable services. Developers can use these services to create applications that perform such tasks as

- creating a conference
- accessing conferences
- reading and writing topics and replies
- searching discussions, using selection criteria

Macintosh users can access VAX Notes conferences by using terminal emulation or by using the VMS DECwindows interface. Third-party solutions also offer a Macintosh client that provides access to VAX Notes conferences with a true Macintosh user interface.

Appendix:

For More Information

This book has described the Apple-Digital network environment. Because the book describes both current and planned capabilities, the availability of individual features and products varies. For information about specific features and products, consult the following Apple and Digital resources:

For information on Apple products, contact these addresses:

Developer Programs
Attn: Apple-Digital Connectivity
Apple Computer, Inc.
20525 Mariani Avenue, MS 75-2C
Cupertino, CA 95014
(408) 974-4897

APDA
Apple Computer, Inc.
20525 Mariani Avenue, MS 33-G
Cupertino, CA 95014
(800) 282-APDA or (800) 282-2732
Fax: 408-562-3971

For information on Digital products, contact your local Digital sales office. Developers may contact this address:

Digital Equipment Corporation
Personal Computing Systems Group, ISV Office
30 Porter Road
Littleton, MA 01460
(508) 486-2511

In addition, the following documentation provides product-specific information:

- ADSP—see the AppleTalk Data Stream Protocol Preliminary Note, AppleTalk Data Stream Protocol Macintosh OS Driver Interface Technical Note, and AppleTalk Data Stream Protocol for VMS Client Interface Specification Release Note, all available through APDA™
- AppleTalk—*AppleTalk Network Overview* (Addison-Wesley Publishing Company), *Inside AppleTalk* (Addison-Wesley Publishing Company and APDA), and *Inside Macintosh,* Volumes II & IV (Addison-Wesley Publishing Company)
- AppleTalk Filing Protocol—see the AppleTalk Filing Protocol (AFP) Engineering Technical Notes (Protocol version 2.0), available through APDA
- AppleTalk for VMS—see APDA*log*™
- Data Access Language and the Data Access Language API—see the Data Access Language (CL/1) *Connectivity Language*, available through APDA
- DDIF—*VMS DECwindows Compound Document Architecture Manual,* and *Digital Technical Journal,* special issue on CDA, 1990, available from Digital Equipment Corporation
- DEC LanWORKS for Macintosh—see *Software Product Description,* information sheet and documentation, available from Digital Equipment Corporation
- DECnet and DNA—see *Digital's Networks: An Architecture with a Future,* available from your local Digital sales office
- DECnet Task-to-Task—see *Digital's Networks: An Architecture with a Future,* available from Digital Equipment Corporation

- Macintosh Communications Toolbox—*The Macintosh Communications Toolbox Reference Guide,* available through APDA, or the Macintosh Communications Toolbox data sheet, available from Apple Computer
- MacTerminal—see the MacTerminal data sheet, available from Apple Computer
- MacWorkStation—*MacWorkStation Programmer's Reference* and *MacWorkStation Programmer's Guide,* available through APDA
- SQL/Services—*Software Product Description* for VAX Rdb/VMS V3.1; *VAX Rdb/VMS* information sheet; *VAX Rdb/VMS, Digital's High-Performance Data Management Solution;* and the VAX Rdb/VMS documentation set (includes *VAX Rdb/VMS Guide to Using SQL/Services*); available from Digital Equipment Corporation
- VAX Notes conferencing—*Guide to VAX Notes,* available from Digital Equipment Corporation
- VAX VTX—see the VAX VTX documentation set (includes *VAX VTX Guide to Designing and Implementing an Infobase, VAX VALU Summary Description,* and *Digital Technical,* Journal No. 6, February 1988, available from Digital Equipment Corporation